

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2019 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОЇ РОБОТИ**

**на тему
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ТРЕНАЖЕРА З ТЕМИ «МЕТОДИ
ПЕРЕВІРКИ ТОТОЖНОЇ ІСТИННОСТІ ЛОГІЧНИХ ФОРМУЛЬ»
ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «МАТЕМАТИЧНА
ЛОГІКА»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Куценко Віталій Володимирович

_____ «___» _____ 2019р.
(підпис)

Науковий керівник к.ф.-м.н., доц. Черненко Оксана Олексіївна

_____ «___» _____ 2019р.
(підпис)

ПОЛТАВА 2019р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	4
1. ПОСТАНОВКА ЗАДАЧІ.....	6
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
2.1. Основні види та призначення платформ дистанційного навчання.....	8
2.2. Огляд математичних тренажерів та їх вади	11
2.3. Актуальність теми роботи.....	20
3. ТЕОРЕТИЧНА ЧАСТИНА	21
3.1. Методи перевірки тотожної істинності логічних формул.....	21
3.2. Приклади застосування	26
3.3. Алгоритмізація за темою роботи.....	29
3.4. Розробка блок-схеми.....	41
4. ПРАКТИЧНА ЧАСТИНА	50
4.1. Обґрунтування вибору програмних засобів.....	50
4.2. Опис процесу програмної реалізації	53
4.3. Опис програми.....	63
4.4. Інструкція по використанню тренажера.....	68
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	73
ДОДАТОК А. КОД ПРОГРАМИ	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
тренажер	навчально-тренувальний пристрій для відпрацювання навичків
дистанційне навчання	засноване на сучасних інформаційних і комунікаційних технологіях навчання й підвищення кваліфікації
виведення у теорії L	послідовність формул $A1, A2, \dots, An$, коли будь-яка з формул Ai є або аксіомою, або безпосереднім наслідком деяких попередніх формул за одним із правил виведення
теорема теорії L	така формула A , коли існує виведення в теорії L формули A , в якому останнім елементом є формула A
NetBeans IDE	середовище розробки
Java	об'єктно-орієнтована мова програмування

ВСТУП

Система дистанційного навчання у поточний час продовжує своє бурхливе поширення.

Бурхливе поширення дистанційного навчання спричинило появу великої кількості навчальних програм і систем, а також освітніх середовищ та платформ дистанційного навчання. Найкращі із них мають стати щоденним робочим інструментом викладача. Навчаючись на дистанційних курсах студент підвищує свій інтелектуальний і творчий потенціал, навчається самостійно приймати рішення.

Мета роботи – розробка елементів програмного забезпечення для тренажера з теми «Методи перевірки тотожної істинності логічних формул» дистанційного навчального курсу «Математична логіка».

Об'єктом розробки в даній роботі є процес дистанційного навчання математичним дисциплінам.

Предметом розробки – програмний продукт, що реалізує тренажер для закріплення знань із застосування методів перевірки тотожної істинності.

Головне завдання – розробити алгоритм роботи тренажера з теми «Методи перевірки тотожної істинності логічних формул» та створити її програмну реалізацію.

Методи розробки – методика застосування методів перевірки тотожної істинності логічних формул. Серед програмного забезпечення – середовище розробки IDE NetBeans, об'єктно-орієнтована мова програмування Java.

Програму можна використовувати при викладанні іноземним студентам теми «Методи перевірки тотожної істинності логічних формул» для перевірки і закріплення знань так, як реалізовано вибір мови між українською та англійською.

Тренажер готовий до використання в дистанційному курсі «Математична логіка».

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі тренажера. У другому розділі описано основні види та призначення платформ дистанційного навчання, огляд математичних тренажерів та їх вади, актуальність теми роботи. У третьому розділі представлено теоретичний матеріал, наведено приклади застосування, описано алгоритм роботи тренажера та представлено його блок-схему. У четвертому розділі – описано обґрунтування вибору програмних засобів, процес програмної реалізації, опис програми і інструкцію по використанню тренажера.

Обсяг пояснювальної записки: 88 стор., в т.ч. основна частина - 65 стор., джерела - 14 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Розглянемо основні завдання роботи:

- розглянути основні види та призначення платформ дистанційного навчання;
- оглянути математичі тренажери та їх вади;
- описати актуальність теми роботи;
- розглянути теоретичні відомості з теми «Методи перевірки тотожної істинності логічних формул»;
- розглянути специфіку застосування методів перевірки тотожної істинності логічних формул;
- розробити алгоритм тренажера, що дозволить закріпити знання і навички;
- розробити тренажер з теми «Методи перевірки тотожної істинності логічних формул»;
- обґрунтувати вибір програмних засобів;
- описати процес програмної реалізації;
- вказати інструкцію по використанню.

Слід розробити можливість студента звернутися до теоретичного матеріалу і прикладів для полегшення проходження тренажера. Також потрібно описати покроково всі дії при розв'язанні прикладів, що ознайомлять студента з методами перевірки тотожної істинності логічних формул та принципом їх застосування.

При невірній відповіді на завдання слід розробити можливість виведення повідомлення про помилку з правильною відповіддю.

Розроблений тренажер повинен містити наступні елементи:

- головну сторінку;
- можливість ознайомитися з теоретичним матеріалом і прикладами;

- сторінку з умовою прикладу та завданням;
- сторінку з повідомленням про ознайомлення з темою.

Щоб програму можна було використовувати при викладанні іноземним студентам теми «Методи перевірки тотожної істинності логічних формул» для перевірки і закріплення знань необхідно реалізувати вибір мови між українською та англійською. При початку проходження тренажера повинно блокуватися вибір мови.

Результати роботи мають бути опубліковані в збірнику наукових статей та матеріалах науково-практичного семінару.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Основні види та призначення платформ дистанційного навчання

Оскільки одним із стратегічних напрямів реформування освітньої системи України є активне використання інформаційних та комунікаційних технологій для розвитку дистанційного навчання необхідно зупинитися на дослідженні застосовування платформ дистанційного навчання, без яких організувати дистанційне навчання неможливо. Вибір платформ дистанційного навчання є дуже важливим кроком.

Світовим лідером серед розробників комерційних продуктів є американська компанія Blackboard Inc. (www.blackboard.com), яка розробила однойменну платформу для електронного навчання "Blackboard". Компанія володіє цілою лінійкою програмних продуктів, які активно використовуються по всьому світу для організації навчального процесу на всіх рівнях освіти. Особливо продукція компанії широко використовується в Північній Америці і Японії. Після придбання іншої великої компанії WebCT, що також спеціалізувалася на електронній освіті, Blackboard зміцнив свої позиції і в Європі. Недоліками цього продукту є висока вартість.

До складу системи Blackboard Learn входять:

- Blackboard Course Delivery - платформа електронного навчання, призначена для управління віртуальним навчальним середовищем і надання платформи для курсів дистанційного навчання;
- Blackboard Content Management - сховище електронних освітніх ресурсів, призначене для централізованого накопичення та структурування електронних освітніх ресурсів, а також управління доступом до них користувачів і зовнішніх додатків;

- Blackboard Community Engagement – навчальний портал, призначений для організації єдиного доступу до сервісів системи Blackboard Learn, забезпечення комунікацій і спільної роботи користувачів.

ATutor є веб-орієнтованою системою керування навчанням (Learning Management System, LMS). Програмний продукт є простим у встановленні, налаштуванні та підтримці для системних адміністраторів; викладачі (інструктори) можуть досить легко створювати та переносити навчальні матеріали та запускати свої онлайн-курси. А оскільки система є модульна, тобто складається з окремих функціональних одиниць — модулів, то вона відкрита для модернізації і розширення функціональних можливостей. Програма розробляється та підтримується з 2001 року Грегом Геєм (Greg Gay), Джоелом Кроненбергом (Joel Kronenberg), Гайді Гейзелтон (Heidi Hazelton) із Дослідницького центру адаптивних технологій Університету Торонто (Adaptive Technology Resource Centre, University of Toronto). Система ATutor поширюється на основі GNU General Public License (GPL), яка, зокрема, дозволяє вільно використовувати, змінювати та доповнювати програму.

Dokeos – платформа побудови сайтів дистанційного навчання, заснована на гілці (fork) Claroline (версії 1.4.2.). Гілка являє собою клон вільно поширюваного програмного продукту, створений з метою змінити додаток-оригінал в тому чи іншому напрямку.

Dokeos безкоштовний і залишиться таким, оскільки ліцензія Claroline (GNU/GPL) припускає, що гілки підпадають під ту ж ліцензію. Оскільки гілка була виділена недавно, обидва додатки зараз відносно схожі один на одного, хоча деякі відмінності в ергономіці, побудові інтерфейсу, функціоналі вже починають проявлятися.

Dokeos – результат роботи деяких членів первісної команди розробників Claroline, які задумали змінити орієнтацію додатку. Справа в тому, що Claroline прекрасно адаптована для університетського

середовища, що виражається в підтримці великої кількості учнів та курсів. Dokeos, більше орієнтований на професійну клієнтуру, наприклад, на персонал підприємства.

Moodle (модульне об'єктно-орієнтоване динамічне навчальне середовище), яке може використовуватися як платформа для електронного, в тому числі дистанційного навчання. Moodle — це безкоштовна, відкрита (Open Source) система управління навчанням. Вона реалізує філософію

«педагогіки соціального конструктивізму» та орієнтована насамперед на організацію взаємодії між викладачем та учнями, хоча підходить і для організації традиційних дистанційних курсів, а також підтримки очного навчання.

Moodle перекладена на десятки мов, в числі й на українську. Система використовується у 175 країнах світу.

Головним розробником системи є Martin Dougiamas з Австралії. Цей проект є відкритим та в ньому бере участь і велика кількість інших розробників.

В сучасному інформаційному суспільстві Moodle набуває все більшого поширення. Сьогодні система використовується не лише в закладах вищої школи, а й загальноосвітніх школах, некомерційних організаціях, приватних компаніях, індивідуальними викладачами і навіть, батьками, що самотійно навчають дітей.

Цьому сприяє те, що система придатна для використання не тільки в варіанті роботи в глобальних мереж, а й легко адаптується під самодостатню платформу для створення локальних одно користувачьких офлайн навчальних ресурсів, та ресурсів, здатних повноцінно функціонувати в рамках локальних мереж. Цей комплекс забезпечує розробника навчального ресурсу великою кількістю інструментів, які надають можливість співпрацювати на рівнях:

- учень – учень;
- учень – викладач;

- викладач – учень в мережному варіанті;
- або учень – система керування курсом;
- система керування курсом – учень в офлайн режимі роботи.

При підготовці та проведенні занять на платформі Moodle викладач може використовувати її можливості, за допомогою яких організовує вивчення матеріалу таким чином, щоб форми навчання відповідали цілям та задачам конкретних занять.

Важливою особливістю Moodle є те, що система створює та зберігає портфоліо кожного слухача: усі роботи, що він зробив, оцінки, коментарі викладача, усі повідомлення на форумі [2, 3].

2.2. Огляд математичних тренажерів та їх вади

До розглянутих тренажерів відносяться тренажери з дисциплін «Математичний аналіз» та «Алгебра і геометрія».

Так в тренажері з теми «Розкриття найпростіших невизначеностей» на головній сторінці виводиться тема, автор-розробник і керівник (рис. 2.1) [4, 5].

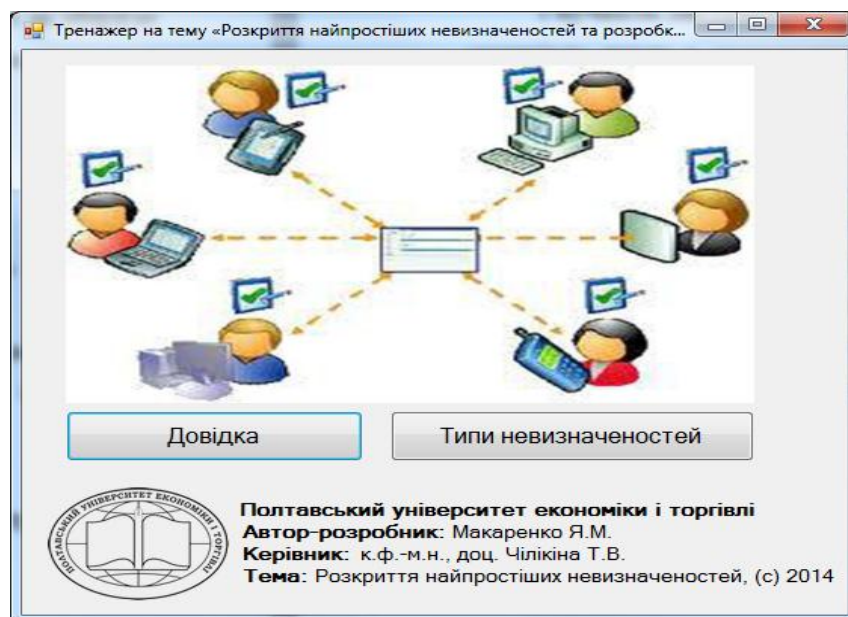


Рисунок 2.1 – Головна сторінка тренажера «Розкриття найпростіших невизначеностей»

Натиснувши кнопку «Довідка» з'явиться вікно з довідкою (рис. 2.2).

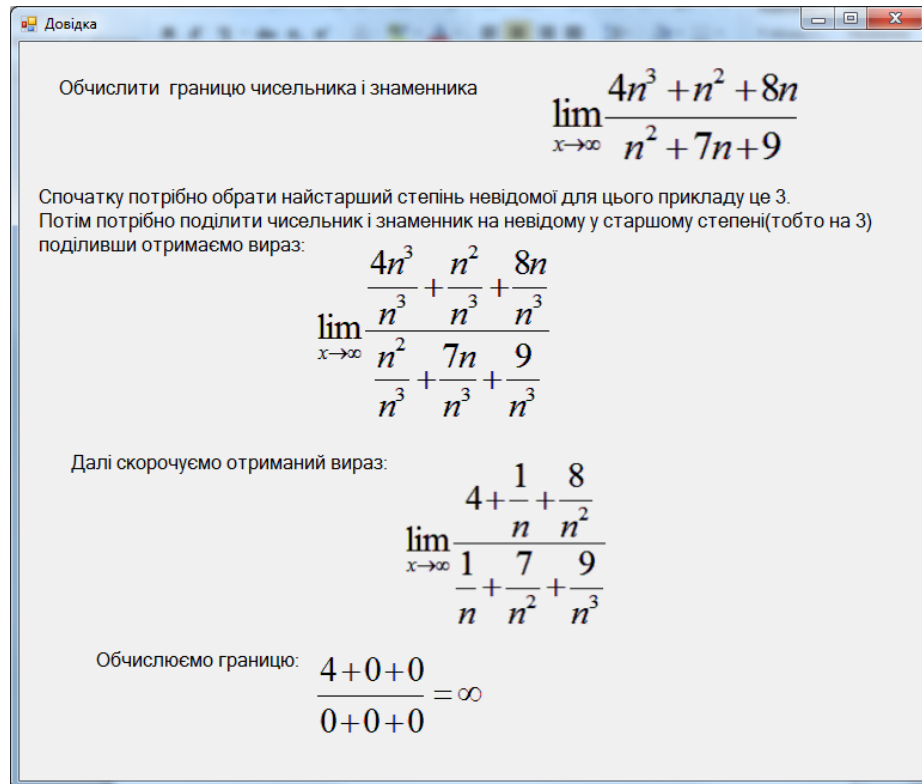


Рисунок 2.2 – Довідка тренажера «Розкриття найпростіших невизначеностей»

Якщо натиснути «Типи невизначеностей» відбудеться перехід до вибору завдання (рис. 2.3).

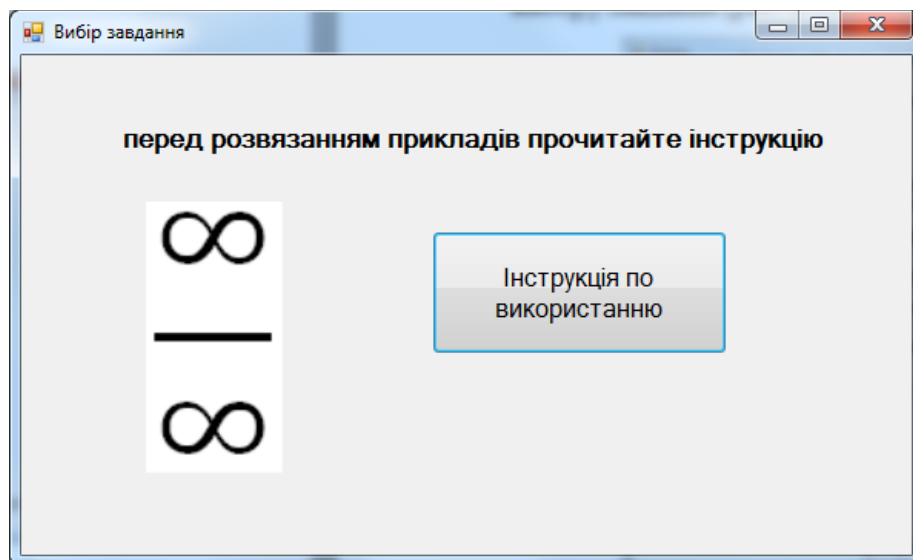


Рисунок 2.3 – Вибір завдання тренажера «Розкриття найпростіших невизначеностей»

На даному кроці можна відкрити інструкцію (рис. 2.4) або виведеться умова прикладу (2.5).

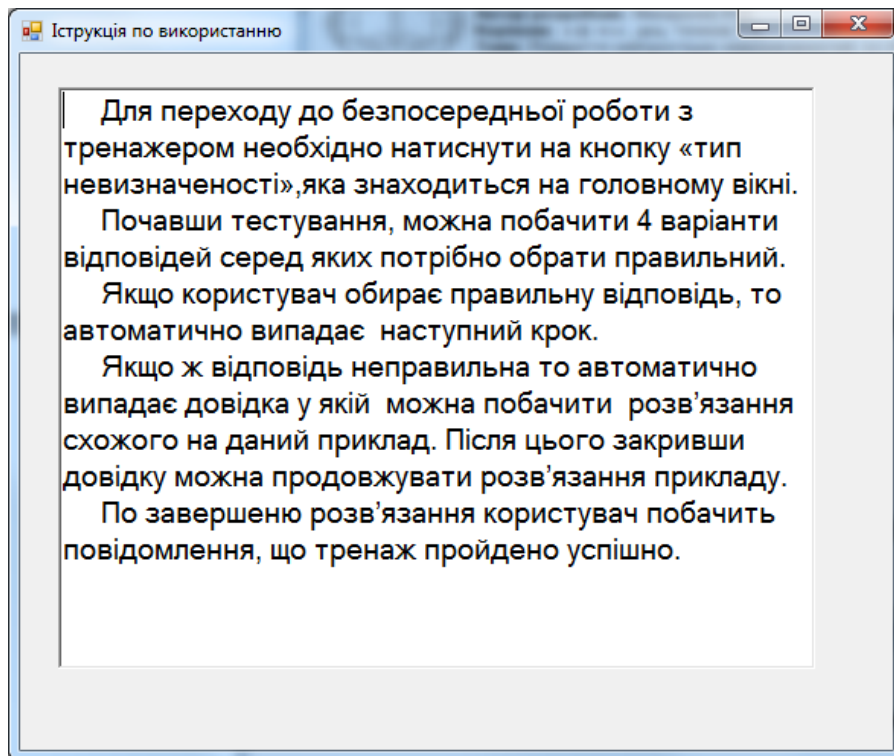


Рисунок 2.4 – Інструкція тренажера «Розкриття найпростіших невизначеностей»

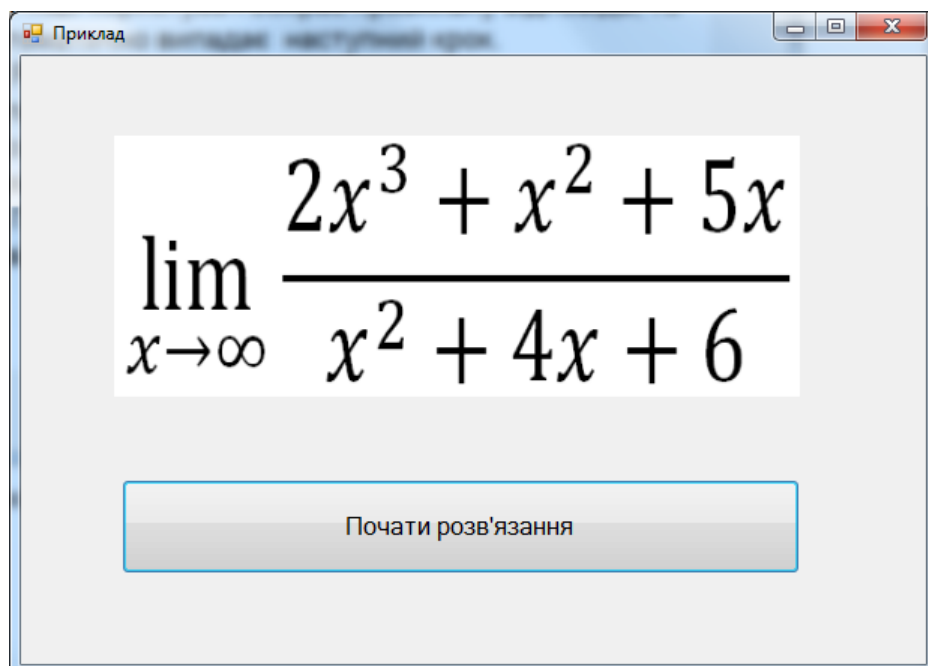


Рисунок 2.5 – Приклад тренажера «Розкриття найпростіших невизначеностей»

У кожному завданні потрібно вибрати відповідь або формулу (рис. 2.6).

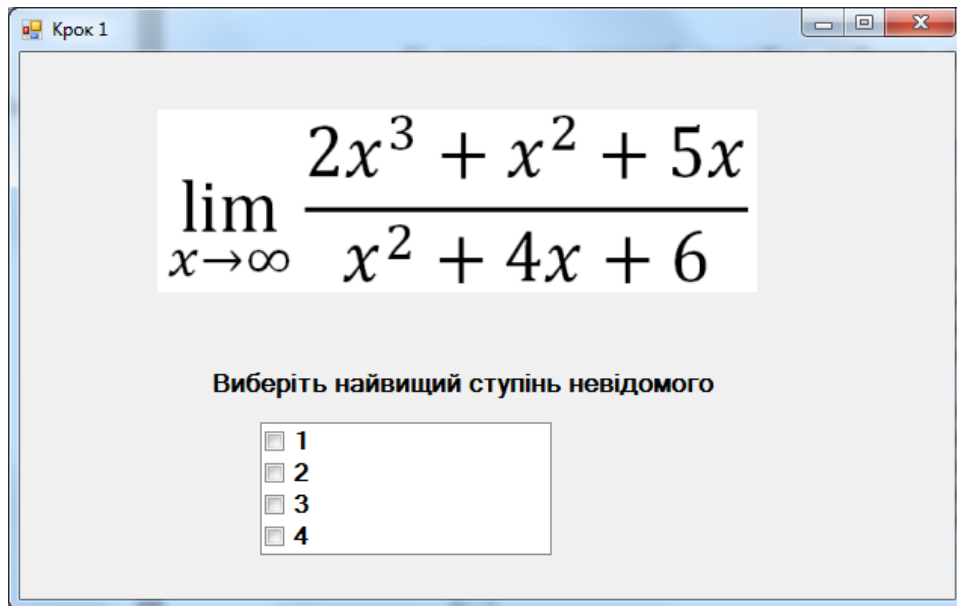


Рисунок 2.6 – Вибір відповіді у тренажері «Розкриття найпростіших невизначеностей»

У разі помилки відображається відповідне повідомлення (рис. 2.7).

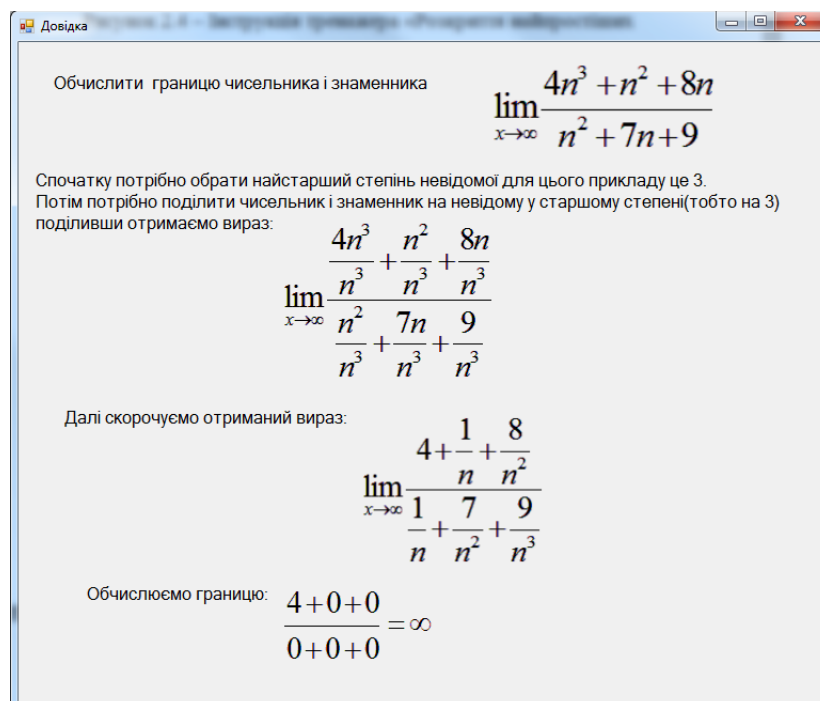


Рисунок 2.7 – Помилка у тренажері «Розкриття найпростіших невизначеностей»

Після проходження тренажера виводиться повідомлення і відбувається перехід на головну сторінку (рис. 2.8).

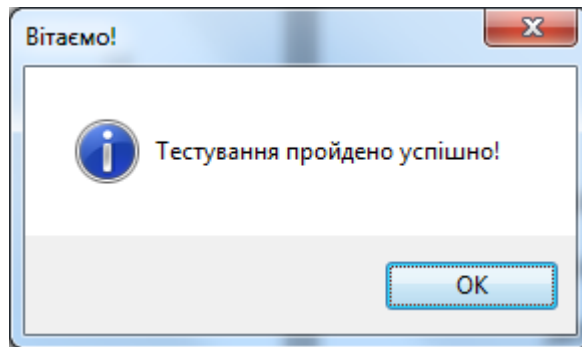


Рисунок 2.8 – Повідомлення про завершення тренажера «Розкриття найпростіших невизначеностей»

Наступний тренажер з теми «Диференціальне числення функції однієї змінної» містить схожу до попереднього загальну інформацію. Також пропонує розглянути теоретичний матеріал або перейти до тренажера (рис. 2.9) [4].

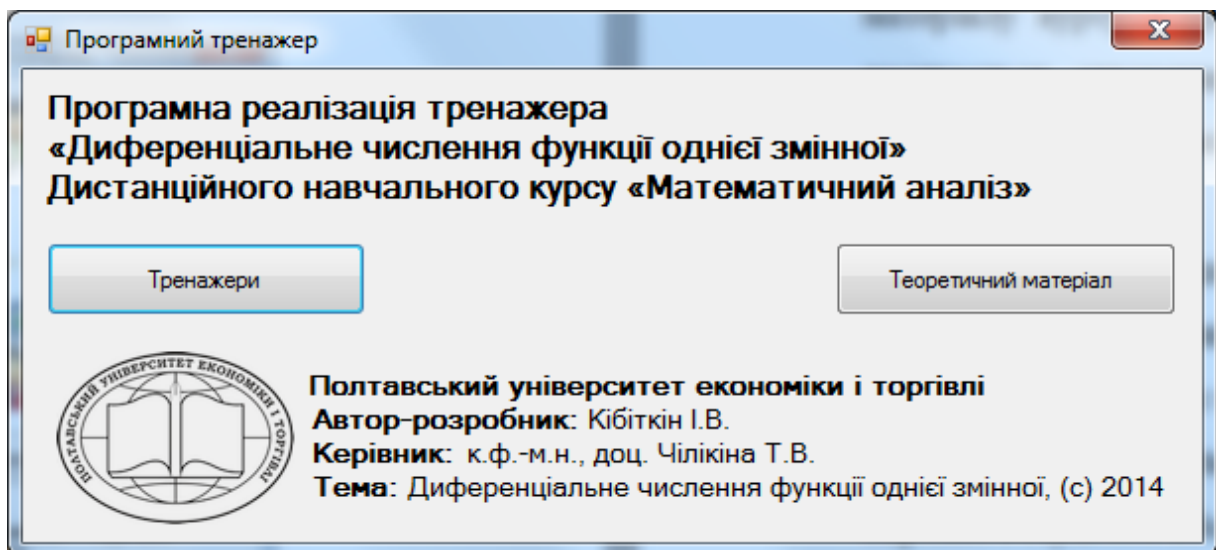


Рисунок 2.9 – Головна сторінка тренажера «Диференціальне числення функції однієї змінної»

В даній програмі користувачу надається можливість запустити вже розроблений тренажер чи створити власний (рис. 2.10).

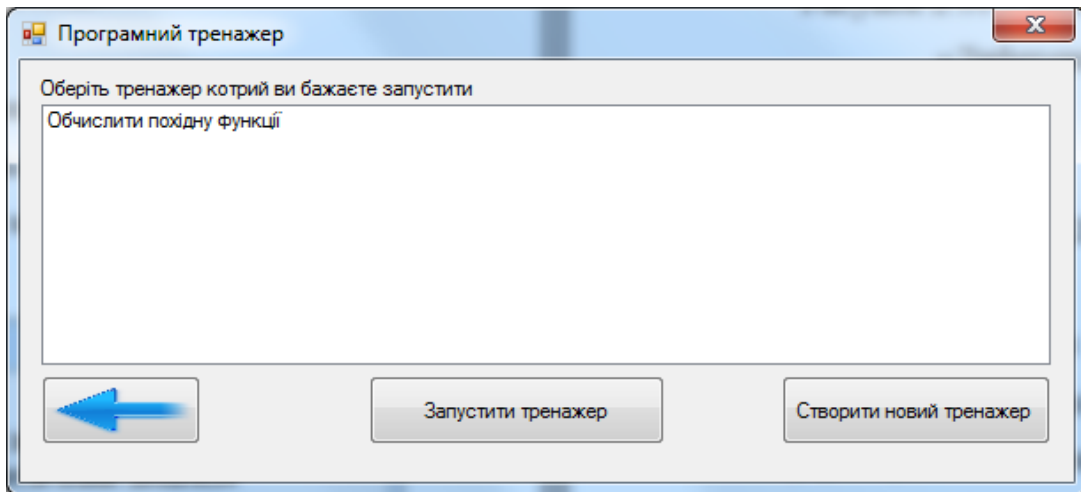


Рисунок 2.10 – Вибір між проходженням чи створенням у тренажері
«Диференціальне числення функції однієї змінної»

Якщо вибрати «Створити новий тренажер», то необхідно вказати його назву і додати запитання (рис. 2.11).

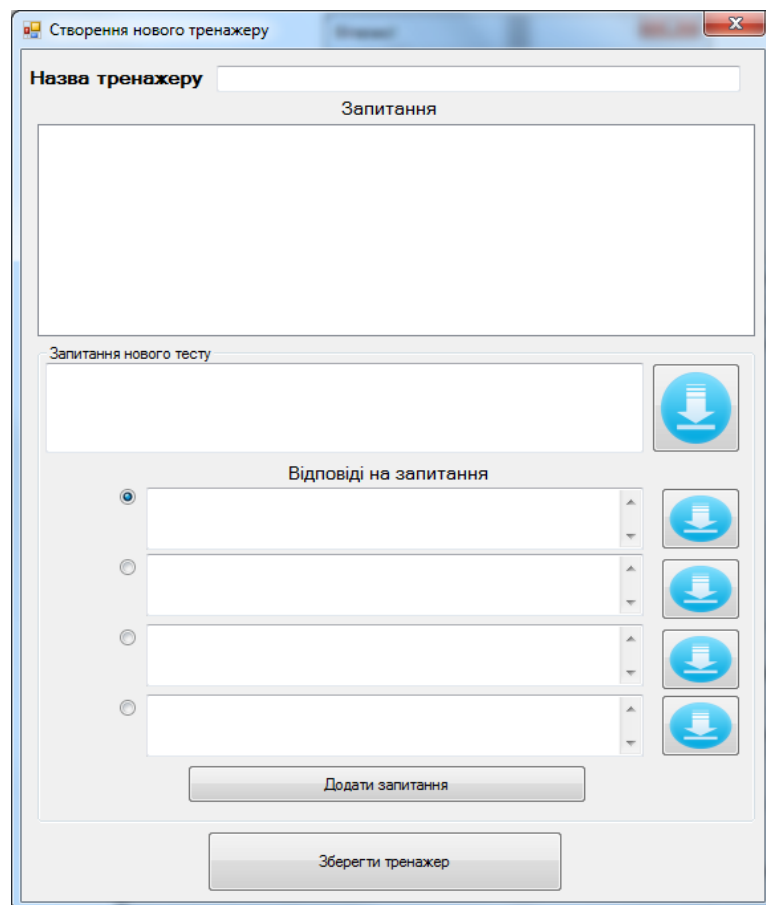


Рисунок 2.11 – Створення нової програми у тренажері «Диференціальне
числення функції однієї змінної»

Після запуску вже існуючого тренажера відображаються питання і варіанти відповідей (рис. 2.12). Спочатку вибирається один з варіантів і перевіряється, а вже потім відбувається перехід до наступного питання.

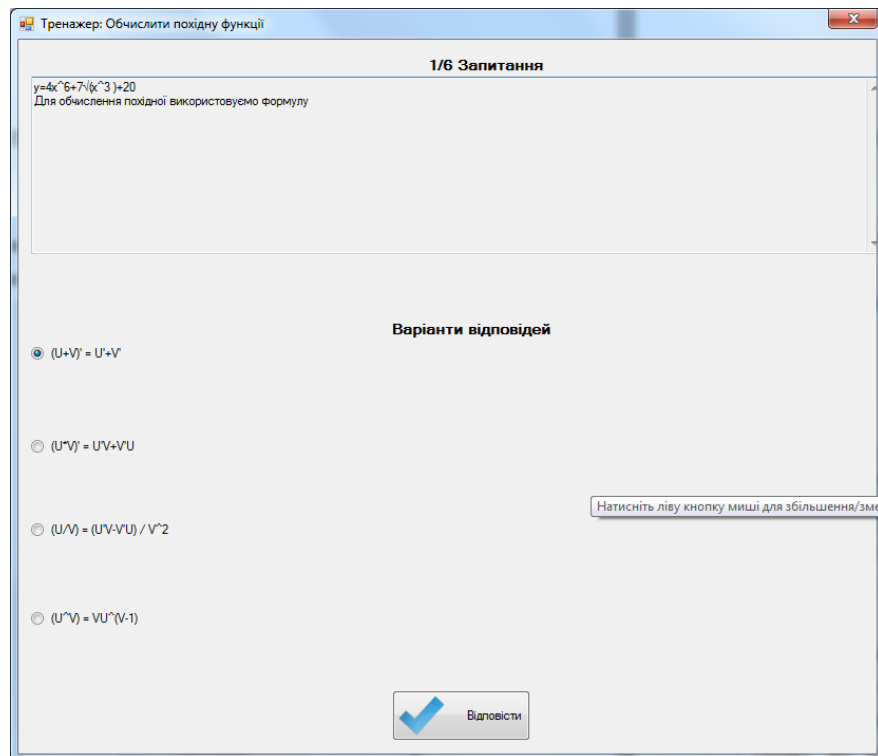


Рисунок 2.12 – Проходження існуючої програми у тренажері
«Диференціальне числення функції однієї змінної»

Після проходження також виводиться відповідне повідомлення і відбувається перехід на головну сторінку.

Останнім було розглянуто тренажер з теми "Матриці і визначники", в якому відразу пропонується перейти до одного з прикладів (рис. 2. 13) [6].

Якщо при виконанні прикладу ввести значення в комірку або вибрати відповідь, то це перевіряється та вказується чи правильно відповіли, чи ні (рис. 2.14).

Також виводиться повідомлення про помилку (рис. 2.15).

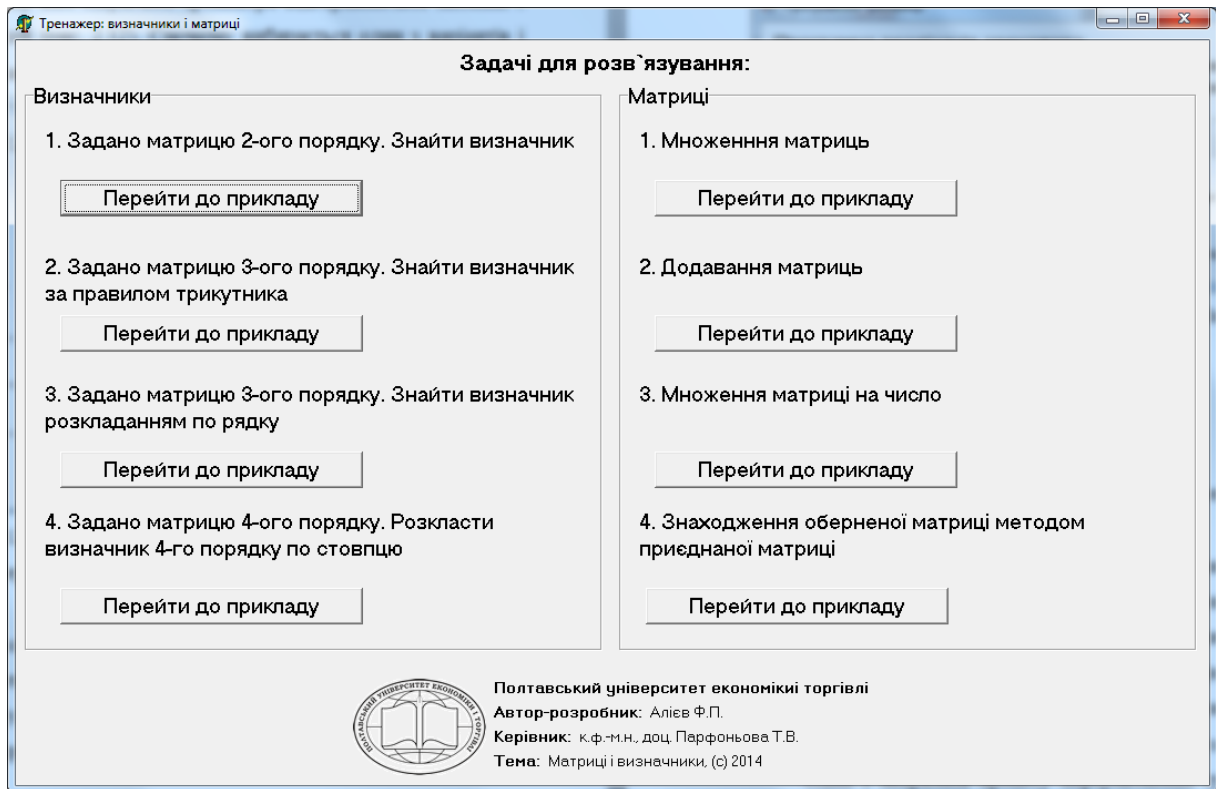


Рисунок 2.13 – Головна сторінка тренажера «Матриці і визначники»

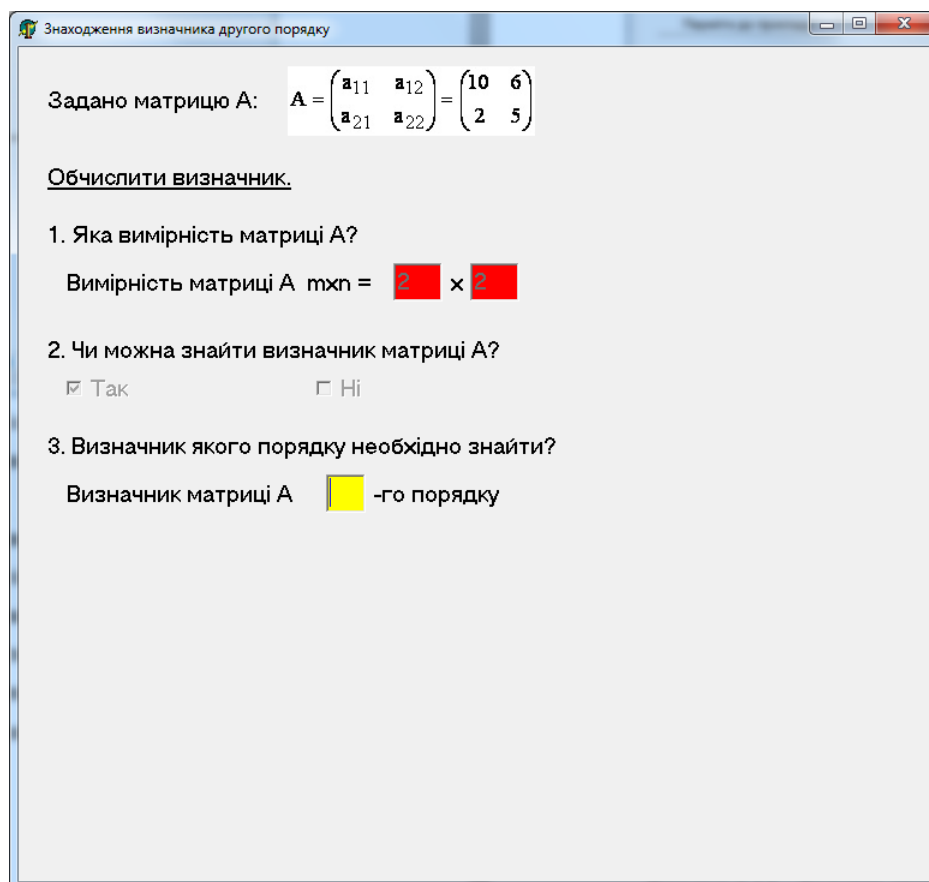


Рисунок 2.14 – Виконання прикладу тренажера «Матриці і визначники»

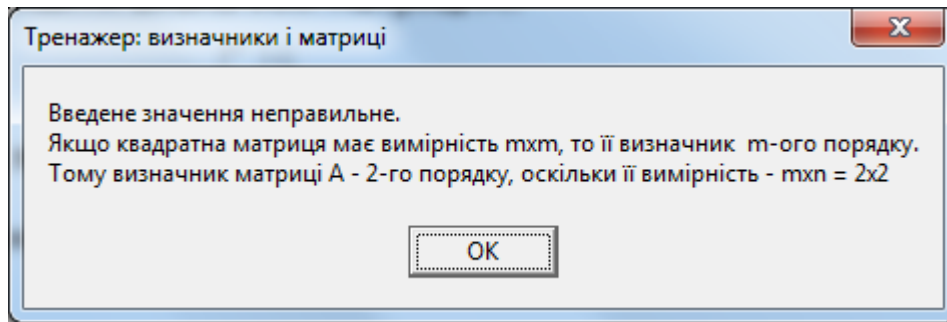


Рисунок 2.15 – Повідомлення про помилку тренажера «Матриці і визначники»

При розв’язанні прикладу пропонується перейти далі або до меню (рис. 2.16).

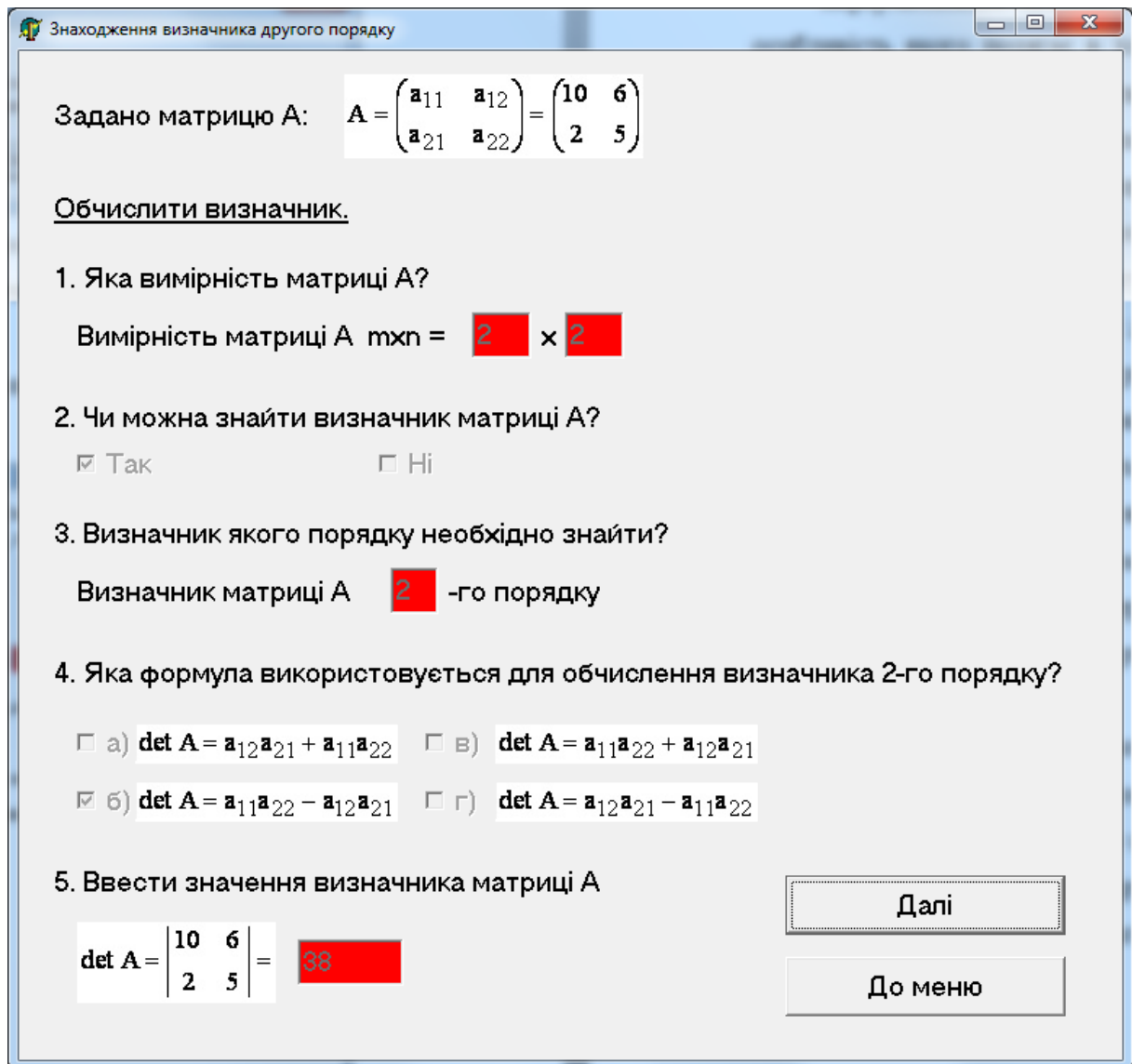


Рисунок 2.16 – Кінець прикладу тренажера «Матриці і визначники»

2.3. Актуальність теми роботи

Інформатизація суспільства – це глобальний соціальний процес, особливість якого полягає в тому, що домінуючим видом діяльності в сфері суспільного виробництва є збір, накопичення, продукування, обробка, зберігання, передача та використання інформації, здійснювані на основі сучасних засобів мікропроцесорної та обчислювальної техніки, а також на базі різноманітних засобів інформаційного обміну.

Один з напрямків процесу інформатизації сучасного суспільства є інформатизація освіти - процес дослідження та забезпечення сфери освіти методологією та практикою розробки та оптимального використання сучасних або, як їх прийнято називати, нових інформаційних технологій, орієнтованих на реалізацію психолого-педагогічних цілей навчання.

Гнучкість дистанційного навчання полягає у можливості викладання матеріалу курсу з урахуванням підготовки та здібностей студентів. Це досягається створенням альтернативних сайтів для одержання більш детальної або додаткової інформації із складних тем, або низки питань-підказок.

Актуальність дистанційного навчання проявляється у можливості упровадження новітніх педагогічних, психологічних і методологічних розробок з розбиттям матеріалу на окремі функціонально завершені модулі (теми), які вивчаються у міру засвоєння і відповідають здібностям окремого студента або групи загалом.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Методи перевірки тотожної істинності логічних формул

При побудові алгебри висловлювань в основу покладено поняття висловлювання, як об'єкту приймаючому логічні значення “Істина” або “Хибність” (закон виключеного третього). Ці поняття в багатьох випадках суб'єктивні і не відносяться до математичних.

В цьому розділі розглядається формалізована аксіоматична теорія – числення висловлювань, яка вільна від указаної вище вимоги і представляє одну з можливих аксіоматизацій алгебри висловлювань.

Довільну формулу F числення висловлювань можна змістовно інтерпретувати як складене висловлювання, істинність або хибність якого залежить від істинності елементарних висловлювань, що до нього входять. Таким чином, кожній формулі F числення висловлювань можна аналогічно тому, як це було зроблено в алгебрі висловлювань, поставити у відповідність функцію істинності f .

При побудові числення висловлювань можуть бути вибрані різні системи аксіом і правила виводу, проте при будь-якому виборі множина формул числення висловлювань збігається з множиною тотожно істинних формул алгебри висловлювань.

Формальна аксіоматична теорія L .

Формальна аксіоматична теорія L є визначеною, якщо виконані такі умови.

1. Задано деякий злічений алфавіт – символи теорії L . Скінченні послідовності символів теорії L називають виразами теорії L .
2. Задано підмножину виразів теорії L , яку називають множиною формул теорії L (часто існує процедура, за допомогою якої можна завжди визначити, чи є даний вираз формулою).

3. Задано деяку множину формул, елементи якої називають аксіомами теорії L (якщо є можливість перевірити, чи є дана формула аксіомою, то теорію L називають ефективно аксіоматизованою, або аксіоматичною теорією).

4. Задано скінченну множину R_1, R_2, \dots, R_j відношень між формулами, які називають правилами виведення.

Для будь-якого $R_i \in J$, таке, що для множини формул A_1, A_2, \dots, A_j і будь-якої формули A ефективно вирішується питання, чи знаходяться ці формули A_1, A_2, \dots, A_j у відношенні R_i з формулою A , і якщо це так, то A називають безпосереднім наслідком формул A_1, A_2, \dots, A_j за правилом $R_i (A_1, A_2, \dots, A_j) \vdash A$.

Виведенням у теорії L називають таку послідовність формул A_1, A_2, \dots, A_n , коли будь-яка з формул A_i є або аксіомою, або безпосереднім наслідком деяких попередніх формул за одним із правил виведення.

Теоремою теорії L називають таку формулу A , коли існує виведення в теорії L формули A , в якому останнім елементом є формула A . Це виведення називають виведенням формули A в теорії L .

Якщо є алгоритм для перевірки, чи A є теоремою теорії L , то теорію L називають розв'язною теорією, інакше – теорією, що не є розв'язною.

Формулу A називають наслідком у теорії L множини формул Γ тоді і тільки тоді, коли існує така послідовність формул A_1, A_2, \dots, A_n , що $A_n = A$, і будь-яка з формул A_i є або аксіомою, або елементом Γ , або безпосереднім наслідком деяких попередніх формул за одним із правил виведення.

Послідовність A_1, A_2, \dots, A_n називають виведенням A із Γ , а елементи множини Γ – гіпотезами виведення.

Теорема 1. Довести, що для будь-якої формули A виконується умова $\vdash LA \rightarrow A \vdash LA \rightarrow A$. Знак \vdash_L - означає, що мова йде про теорію L .

Теорема дедукції.

Теорема дедукції дає нове правило виведення у логіці висловлювань, і тому вона є важливою при доведенні теорем і різних висловлювань.

Теорема дедукції. Якщо Γ – множина формул, A і B – формули і Γ , $A \vdash B$, то $\Gamma \vdash A \rightarrow B$. Зокрема, якщо $A \vdash B$, то $\vdash A \rightarrow B$.

Теорема 2. Довести, що $A \rightarrow B, B \rightarrow C \vdash A \rightarrow C$.

Теорема 3. Довести, що $A \rightarrow (B \rightarrow C), B \vdash A \rightarrow C$

Теорема 4. Для будь-яких формул A і B такі формули є теоремами формальної теорії L :

- | | |
|--|---|
| а) $\neg\neg B \rightarrow B$; | д) $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$; |
| б) $B \rightarrow \neg\neg B$; | е) $A \rightarrow (\neg B \rightarrow \neg(A \rightarrow B))$; |
| в) $\neg A \rightarrow (A \rightarrow B)$; | ж) $(A \rightarrow B) \rightarrow ((\neg A \rightarrow B) \rightarrow B)$. |
| г) $(\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$; | |
| з) $\vdash \neg\neg B \rightarrow B$ | |

Отже,

$$A \rightarrow B, \neg A \rightarrow B \vdash B \quad A \rightarrow B, \neg A \rightarrow B \vdash B,$$

і застосувавши двічі теорему дедукції, отримаємо

$$\vdash (A \rightarrow B) \rightarrow ((\neg A \rightarrow B) \rightarrow B) \vdash (A \rightarrow B) \rightarrow ((\neg A \rightarrow B) \rightarrow B).$$

Аксіоматичний метод.

В численні висловлень визначають такі схеми аксіом:

1. $a \rightarrow (b \rightarrow a)$
2. $(a \rightarrow b) \rightarrow ((a \rightarrow (b \rightarrow c)) \rightarrow (a \rightarrow c))$
3. $(a \wedge b) \rightarrow a$
4. $(a \wedge b) \rightarrow b$
5. $(a \rightarrow b) \rightarrow ((a \rightarrow c) \rightarrow (a \rightarrow (b \wedge c)))$
6. $a \rightarrow (a \vee b)$
7. $b \rightarrow (a \vee b)$

$$8. (a \rightarrow c) \rightarrow ((b \rightarrow c) \rightarrow ((a \vee b) \rightarrow c))$$

$$9. (a \rightarrow \neg b) \rightarrow (b \rightarrow \neg a)$$

$$10. \neg\neg a \rightarrow a$$

Єдиним правилом виводу є:

$$\frac{a \rightarrow b, a}{b} \text{ (Modus ponens)}$$

Конструктивний метод.

Протилежним до аксіоматичного є конструктивний метод у логіці висловлювань, який ґрунтується на таблицях істинності.

В математиці конструктивне доведення — це метод доведення, що підтверджує існування математичного об'єкта шляхом надання або створення способу відтворення даного об'єкта. Він протиставляється неконструктивному доведенню (також відомому як теорема доведення існування, або чиста теорема існування), яке доводить існування певного об'єкта без надання прикладів.

Деякі неконструктивні доведення показують, що якщо якась передумова помилкова, то впливає протиріччя; отже, передумова повинна бути істинною (доведення від протилежного). Проте в деяких розділах конструктивної математики, в тому числі в інтуїціонізмі, був прийнятий принцип вибуху (лат. *ex Falso QuodLibet*, «за брехнею нічого не слідує»).

Конструктивізм — це математична філософія, яка відкидає всі, за виключенням конструктивних, докази в математиці. Це призводить до обмеження дозволених методів доведень (спочатку закон виключеного третього не було прийнято застосовувати) і іншого змісту термінології (наприклад, термін «або» має більш обмежений зміст в конструктивній математиці, ніж в класичній).

Конструктивні доведення можна розглядати як визначення перевірених математичних алгоритмів: ця ідея досліджується в інтерпретації конструктивної логіки Брауера-Гейтінга-Колмогорова, у

відповідності Каррі-Говарда, а також в таких логічних системах, як інтуїціоністська теорія типів Пера Мартіна-Льофа та числення конструкцій Тьєррі Кокана й Жерара Гуета.

Метод доведення від супротивного.

Цей метод доведення істинності висловлювання полягає в такому: допускають, що істинним є заперечення того висловлювання, яке необхідно довести (наслідок); потім через причини (посилки) намагаються дійти до суперечності. Якщо це відбулося, то досліджуване логічне висловлювання істинне, якщо ні – хибне.

Доведення від супротивного (зведення до абсурду, лат. *Reductio ad absurdum*) — один із поширених методів доведення тверджень в математичній логіці. Доведення від супротивного — вид доведення, при якому доведення деякого твердження відбувається через спростування заперечення цього твердження — антитезису. Метод ґрунтується на правильності формули $((A \rightarrow B) \wedge \neg B) \rightarrow \neg A$ в численні висловлень та законі подвійного заперечення.

Припускаємо, що A є істинним твердженням, і доводимо, що, по-перше, з A виводиться B , а по-друге, що з A виводиться $\neg B$, що неможливо; отже, A хибне, тобто істинне $\neg A$.

Доведення твердження A проводиться наступним чином. Спочатку приймають припущення, що твердження A невірне, а потім доводять, що при такому припущенні було б вірним деяке твердження B , яке завчасно невірне. Отримане протиріччя показує, що початкове твердження було невірним, і тому вірним є твердження $\neg\neg A$, яке по закону подвійного заперечення дорівнює твердженню A .

Метод резолюцій.

У цьому методі для доведення істинності висловлювань використовують аксіоми порядку. Суть методу зводиться до того, що дві посилки диз'юнкти з протилежними термами завжди можливо склеїти в один остаточний диз'юнкт, у якому вже не буде протилежних термів.

Метод резолюцій за своєю суттю заміняє аксіому порядку, оскільки вона сама може бути доведена в рамках методу резолюцій [7, 8].

3.2. Приклади застосування

1. Формули і підстановки.

Приклад 1. Нехай A, B, C – формули числення висловлювань.

Тоді $(C \rightarrow (A \vee B))$, $((\bar{A}) \wedge B) \rightarrow (\bar{C})$ також формули.

Приклад 2. Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$

Розв'язання. Розбиття дужок на пари здійснюється індексами дужок:

$$({}_1({}_2A \rightarrow B_2) \rightarrow ({}_3({}_4A \rightarrow C_4) \rightarrow ({}_5A \rightarrow ({}_6B \wedge C_6)_5)_3)_1).$$

Область дії оператора вказано парою дужок:

- область дії першого оператора – це дужки 2-2, $A \rightarrow B$;
- другого – дужки 1-1, $(A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C)))$;
- третього – дужки 4-4, $A \rightarrow C$;
- четвертого – дужки 3-3, $(A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))$;
- п'ятого – дужки 5-5, $A \rightarrow (B \wedge C)$;
- шостого – дужки 6-6, $B \wedge C$.

Приклад 3. Чи є формулами числення висловлень такі вирази:

- 1) $((A \vee ((\neg B) \rightarrow C)) \rightarrow (\neg A))$;
- 2) $((A \wedge B) \vee (C \wedge D) \rightarrow (A \wedge C))$;
- 3) $(A \rightarrow C) \leftrightarrow ((\neg C) \rightarrow (\neg A))$;
- 4) $((A \wedge B) \wedge C) \rightarrow ((B \vee C) \vee (\neg D))$.

Розв'язання. Формулами числення висловлень є вирази 1, 2, 4. Вираз 3 не є формулою, оскільки не має відповідності дужок.

Приклад 4. Виконати підстановки:

$$1) S_{B \rightarrow A}^A A \rightarrow (B \rightarrow A);$$

$$2) S_{B \rightarrow A}^{\neg A} (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A);$$

$$3) S_C^{B \rightarrow C} (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)).$$

Розв'язання. За означенням підстановки, маємо:

$$1) S_{B \rightarrow A}^A A \rightarrow (B \rightarrow A) = (B \rightarrow A) \rightarrow (B \rightarrow (B \rightarrow A));$$

$$2) S_{B \rightarrow A}^{\neg A} (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A) = (A \rightarrow B) \rightarrow (\neg B \rightarrow (B \rightarrow A));$$

$$3) S_C^{B \rightarrow C} (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) = \\ = (A \rightarrow C) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)).$$

2. Формальні доведення та вивідність формул.

У якості аксіом числення висловлень візьмемо такі формули:

$$A1. a \rightarrow (b \rightarrow a)$$

$$A2. (a \rightarrow b) \rightarrow ((a \rightarrow (b \rightarrow c)) \rightarrow (a \rightarrow c))$$

$$A3. (a \wedge b) \rightarrow a$$

$$A4. (a \wedge b) \rightarrow b$$

$$A5. (a \rightarrow b) \rightarrow ((a \rightarrow c) \rightarrow (a \rightarrow (b \wedge c)))$$

$$A6. a \rightarrow (a \vee b)$$

$$A7. b \rightarrow (a \vee b)$$

$$A8. (a \rightarrow c) \rightarrow ((b \rightarrow c) \rightarrow ((a \vee b) \rightarrow c))$$

$$A9. (a \rightarrow \neg b) \rightarrow (b \rightarrow \neg a)$$

$$A10. \neg \neg a \rightarrow a$$

У якості правил виведення беремо правило підстановки і правило висновку (modus ponens, скорочено – MP).

Приклад 5. Довести, що $\neg(a \rightarrow a)$.

Розв'язання. Формальне доведення таке:

$$F1: S_{b \rightarrow a, a}^{b, c} A2 = (a \rightarrow (b \rightarrow a)) \rightarrow ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a));$$

$$F2: MP(A1, F1) = ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a));$$

$$F3: S_{b \rightarrow a}^b A1 = (a \rightarrow ((b \rightarrow a) \rightarrow a));$$

$$F4: MP(F3, F2) = (a \rightarrow a).$$

Приклад 6. Довести, що $a \neg(b \rightarrow a)$.

Розв'язання. Формальна вивідність така:

$$F1: a;$$

$$F2: MP(F1, A1) = (b \rightarrow a).$$

Приклад 7. Довести, що $a, b, a \rightarrow (b \rightarrow c) \neg c$.

Розв'язання. Формальна вивідність така:

$$F1: a;$$

$$F2: b;$$

$$F3: a \rightarrow (b \rightarrow c);$$

$$F4: MP(F1, F3) = b \rightarrow c;$$

$$F5: MP(F2, F4) = c.$$

Приклад 8. Довести, що $a, b \neg(a \wedge b)$.

Розв'язання. Формальна вивідність така:

$$F1: S_{a, b}^{b, c} A5 = ((a \rightarrow a) \rightarrow ((a \rightarrow b) \rightarrow (a \rightarrow (a \wedge b))));$$

$$F2: (a \rightarrow a) \text{ (див. приклад 5);}$$

$$F3: MP(F2, F1) = ((a \rightarrow b) \rightarrow (a \rightarrow (a \wedge b)));$$

$F4: b;$

$F5: S_{b,a}^{a,b} A1 = (b \rightarrow (a \rightarrow b));$

$F6: MP(F4, F5) = (a \rightarrow b);$

$F7: MP(F6, F3) = (a \rightarrow (a \wedge b));$

$F8: a;$

$F9: MP(F8, F7) = (a \wedge b) [9].$

3.3. Алгоритмізація за темою роботи

Користувачу на головній сторінці тренажера відображається назва цього тренажера, інформація про розробника і керівника, надається можливість ознайомитися з теоретичним матеріалом і прикладами.

Тренажер містить два блоки прикладів:

- Формули і підстановки;
- Формальні доведення та вивідність формул.

1. Формули і підстановки.

Крок 1. Користувачу виводиться умова: «Нехай A, B, C – формули числення висловлювань» і завдання: «Чи є $(C \rightarrow (A \vee B))$ також формулою?». Виводяться варіанти відповіді:

- Так, $(C \rightarrow (A \vee B))$ також є формулою.
- Ні, $(C \rightarrow (A \vee B))$ не є формулою.

Якщо відповідь правильна, тобто вибрано перший варіант, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! $(C \rightarrow (A \vee B))$ також є формулою».

Крок 2. Користувачу виводиться умова: «Нехай A, B, C – формули числення висловлювань» і завдання: «Чи є $((\bar{A}) \wedge B) \rightarrow (\bar{C})$ також формулою?». Виводяться варіанти відповіді:

- Так, $\left(\left(\left(\bar{A}\right) \wedge B\right) \rightarrow\left(\bar{C}\right)\right)$ також є формулою.
- Ні, $\left(\left(\left(\bar{A}\right) \wedge B\right) \rightarrow\left(\bar{C}\right)\right)$ не є формулою.

Якщо відповідь правильна, тобто вибрано перший варіант, то відбувається перехід на крок 3 до наступного прикладу, в іншому разі – відображається повідомлення про помилку: «Помилка! $\left(\left(\left(\bar{A}\right) \wedge B\right) \rightarrow\left(\bar{C}\right)\right)$ також є формулою».

Крок 3. Користувачу виводиться умова: «Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $\left(\left(A \rightarrow B\right) \rightarrow\left(\left(A \rightarrow C\right) \rightarrow\left(A \rightarrow\left(B \wedge C\right)\right)\right)\right)$ » і завдання: «Розбити дужки на пари». Виводиться формула, де потрібно вписати номери дужок у відповідні комірки (рис. 3.1).

$$\left(\square\left(\square A \rightarrow B \square\right) \rightarrow\left(\square\left(\square A \rightarrow C \square\right) \rightarrow\left(\square A \rightarrow\left(\square B \wedge C \square\right) \square\right) \square\right) \square\right)$$

Рисунок 3.1 – Схематичне представлення розбиття дужок

Якщо заповнено правильно всі комірки, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! Правильне розбиття дужок:

$$\left({}_1\left({}_2 A \rightarrow B_2\right) \rightarrow\left({}_3\left({}_4 A \rightarrow C_4\right) \rightarrow\left({}_5 A \rightarrow\left({}_6 B \wedge C_6\right)_5\right)_3\right)_1\right)$$

».

Крок 4. Користувачу виводиться умова: «Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $\left(\left(A \rightarrow B\right) \rightarrow\left(\left(A \rightarrow C\right) \rightarrow\left(A \rightarrow\left(B \wedge C\right)\right)\right)\right)$ » і завдання: «Вкажіть область дії першого оператора». Виводяться варіанти відповіді:

- дужки 2-2, $A \rightarrow B$;
- дужки 1-1, $\left(A \rightarrow B\right) \rightarrow\left(\left(A \rightarrow C\right) \rightarrow\left(A \rightarrow\left(B \wedge C\right)\right)\right)$;
- дужки 3-3, $\left(A \rightarrow C\right) \rightarrow\left(A \rightarrow\left(B \wedge C\right)\right)$;

- дужки 6-6, $B \wedge C$.

Якщо відповідь правильна, тобто вибрано перший варіант, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! Область дії першого оператора це дужки 2-2, $A \rightarrow B$ ».

Крок 5. Користувачу виводиться умова: «Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$ » і завдання: «Вкажіть область дії другого оператора». Виводяться варіанти відповіді:

- дужки 2-2, $A \rightarrow B$;
- дужки 1-1, $(A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C)))$;
- дужки 3-3, $(A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))$;
- дужки 6-6, $B \wedge C$.

Якщо відповідь правильна, тобто вибрано другий варіант, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! Область дії другого оператора це дужки 1-1, $(A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C)))$ ».

Крок 6. Користувачу виводиться умова: «Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$ » і завдання: «Вкажіть область дії третього оператора». Виводяться варіанти відповіді:

- дужки 4-4, $A \rightarrow C$;
- дужки 5-5, $A \rightarrow (B \wedge C)$;
- дужки 3-3, $(A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))$;
- дужки 6-6, $B \wedge C$.

Якщо відповідь правильна, тобто вибрано перший варіант, то відбувається перехід на наступний крок, в іншому разі – відображається

повідомлення про помилку: «Помилка! Область дії третього оператора це дужки 4-4, $A \rightarrow C$ ».

Крок 7. Користувачу виводиться умова: «Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$ » і завдання: «Вкажіть область дії четвертого оператора». Виводяться варіанти відповіді:

- дужки 4-4, $A \rightarrow C$;
- дужки 5-5, $A \rightarrow (B \wedge C)$;
- дужки 3-3, $(A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))$;
- дужки 6-6, $B \wedge C$.

Якщо відповідь правильна, тобто вибрано третій варіант, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! Область дії четвертого оператора це дужки 3-3, $(A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))$ ».

Крок 8. Користувачу виводиться умова: «Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$ » і завдання: «Вкажіть область дії п'ятого оператора». Виводяться варіанти відповіді:

- дужки 4-4, $A \rightarrow C$;
- дужки 5-5, $A \rightarrow (B \wedge C)$;
- дужки 3-3, $(A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))$;
- дужки 6-6, $B \wedge C$.

Якщо відповідь правильна, тобто вибрано другий варіант, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! Область дії п'ятого оператора це дужки 5-5, $A \rightarrow (B \wedge C)$ ».

Крок 9. Користувачу виводиться умова: «Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$ » і завдання: «Вкажіть область дії шостого оператора». Виводяться варіанти відповіді:

- дужки 4-4, $A \rightarrow C$;
- дужки 5-5, $A \rightarrow (B \wedge C)$;
- дужки 3-3, $(A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))$;
- дужки 6-6, $B \wedge C$.

Якщо відповідь правильна, тобто вибрано четвертий варіант, то відбувається перехід на крок 10 до наступного прикладу, в іншому разі – відображається повідомлення про помилку: «Помилка! Область дії шостого оператора це дужки 6-6, $B \wedge C$ ».

Крок 10. Користувачу виводиться завдання: «Чи є формулою числення висловлень вираз $((A \vee ((\neg B) \rightarrow C)) \rightarrow (\neg A))$?». Виводяться варіанти відповіді:

- Так, $((A \vee ((\neg B) \rightarrow C)) \rightarrow (\neg A))$ є формулою.
- Ні, $((A \vee ((\neg B) \rightarrow C)) \rightarrow (\neg A))$ не є формулою.

Якщо відповідь правильна, тобто вибрано перший варіант, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! $((A \vee ((\neg B) \rightarrow C)) \rightarrow (\neg A))$ є формулою числення висловлень».

Крок 11. Користувачу виводиться завдання: «Чи є формулою числення висловлень вираз $((A \wedge B) \vee (C \wedge D) \rightarrow (A \wedge C))$?». Виводяться варіанти відповіді:

- Так, $((A \wedge B) \vee (C \wedge D) \rightarrow (A \wedge C))$ є формулою.
- Ні, $((A \wedge B) \vee (C \wedge D) \rightarrow (A \wedge C))$ не є формулою.

Якщо відповідь правильна, тобто вибрано перший варіант, то відбувається перехід на наступний крок, в іншому разі – відображається

повідомлення про помилку: «Помилка! $((A \wedge B) \vee (C \wedge D) \rightarrow (A \wedge C))$ є формулою числення висловлень».

Крок 12. Користувачу виводиться завдання: «Чи є формулою числення висловлень вираз $(A \rightarrow C) \leftrightarrow ((\neg C) \rightarrow (\neg A))$?». Виводяться варіанти відповіді:

- Так, $(A \rightarrow C) \leftrightarrow ((\neg C) \rightarrow (\neg A))$ є формулою.
- Ні, $(A \rightarrow C) \leftrightarrow ((\neg C) \rightarrow (\neg A))$ не є формулою.

Якщо відповідь правильна, тобто вибрано другий варіант, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! $(A \rightarrow C) \leftrightarrow ((\neg C) \rightarrow (\neg A))$ не є формулою числення висловлень».

Крок 13. Користувачу виводиться завдання: «Чи є формулою числення висловлень вираз $((A \wedge B) \wedge C) \rightarrow ((B \vee C) \vee (\neg D))$?». Виводяться варіанти відповіді:

- Так, $((A \wedge B) \wedge C) \rightarrow ((B \vee C) \vee (\neg D))$ є формулою.
- Ні, $((A \wedge B) \wedge C) \rightarrow ((B \vee C) \vee (\neg D))$ не є формулою.

Якщо відповідь правильна, тобто вибрано перший варіант, то відбувається перехід на крок 14 до наступного прикладу, в іншому разі – відображається повідомлення про помилку: «Помилка! $((A \wedge B) \wedge C) \rightarrow ((B \vee C) \vee (\neg D))$ є формулою числення висловлень».

Крок 14. Користувачу виводиться умова: «Виконати підстановку $S_{B \rightarrow A}^A A \rightarrow (B \rightarrow A)$ » і завдання: «Вкажіть результат підстановки». Виводяться варіанти відповіді:

- $A \rightarrow (B \rightarrow (B \rightarrow A))$;
- $(B \rightarrow A) \rightarrow (B \rightarrow A)$;
- $(B \rightarrow A) \rightarrow (B \rightarrow (B \rightarrow A))$;
- $(A \rightarrow B) \rightarrow (B \rightarrow (A \rightarrow B))$.

Якщо відповідь правильна, тобто вибрано третій варіант, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! За означенням підстановки, маємо

$$S_{B \rightarrow A}^A A \rightarrow (B \rightarrow A) = (B \rightarrow A) \rightarrow (B \rightarrow (B \rightarrow A))$$

».

Крок 15. Користувачу виводиться умова: «Виконати підстановку $S_{B \rightarrow A}^{-A} (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$ » і завдання: «Вкажіть результат підстановки».

Виводяться варіанти відповіді:

- $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$;
- $((B \rightarrow A) \rightarrow B) \rightarrow (\neg B \rightarrow (B \rightarrow A))$;
- $(A \rightarrow B) \rightarrow (\neg B \rightarrow (A \rightarrow B))$;
- $(A \rightarrow B) \rightarrow (\neg B \rightarrow (B \rightarrow A))$.

Якщо відповідь правильна, тобто вибрано четвертий варіант, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! За означенням підстановки, маємо

$$S_{B \rightarrow A}^{-A} (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A) = (A \rightarrow B) \rightarrow (\neg B \rightarrow (B \rightarrow A))$$

».

Крок 16. Користувачу виводиться умова: «Виконати підстановку $S_C^{B \rightarrow C} (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$ » і завдання: «Вкажіть результат підстановки». Виводяться варіанти відповіді:

- $(A \rightarrow (B \rightarrow (B \rightarrow C))) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow (B \rightarrow C)))$;
- $(A \rightarrow C) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$;
- $(A \rightarrow (B \rightarrow (B \rightarrow C))) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$;
- $(A \rightarrow C) \rightarrow ((A \rightarrow B) \rightarrow C)$.

Якщо відповідь правильна, тобто вибрано другий варіант, то відбувається перехід на крок 17 до результатів, в іншому разі –

відображається повідомлення про помилку: «Помилка! За означенням підстановки, маємо

$$\begin{aligned} S_c^{B \rightarrow C} (A \rightarrow (B \rightarrow C)) &\rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) = \\ &= (A \rightarrow C) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) \end{aligned}$$

».

Крок 17. Користувачу виводиться повідомлення про завершення: «Вітаємо! Ви ознайомилися з темою «Формули і підстановки.». Пропонується перейти до розв’язування прикладів наступного блоку або завершити роботу тренажера.

2. Формальні доведення та вивідність формул.

Відображається умова: «У якості аксіом числення висловлень візьмемо такі формули:

$$A1. a \rightarrow (b \rightarrow a)$$

$$A2. (a \rightarrow b) \rightarrow ((a \rightarrow (b \rightarrow c)) \rightarrow (a \rightarrow c))$$

$$A3. (a \wedge b) \rightarrow a$$

$$A4. (a \wedge b) \rightarrow b$$

$$A5. (a \rightarrow b) \rightarrow ((a \rightarrow c) \rightarrow (a \rightarrow (b \wedge c)))$$

$$A6. a \rightarrow (a \vee b)$$

$$A7. b \rightarrow (a \vee b)$$

$$A8. (a \rightarrow c) \rightarrow ((b \rightarrow c) \rightarrow ((a \vee b) \rightarrow c))$$

$$A9. (a \rightarrow \neg b) \rightarrow (b \rightarrow \neg a)$$

$$A10. \neg \neg a \rightarrow a$$

У якості правил виведення беремо правило підстановки і правило висновку (modus ponens, скорочено – MP).»

Крок 1. Користувачу виводиться завдання: «Довести, що $\neg(a \rightarrow a)$. Встановіть послідовність формального доведення. Вкажіть перший крок».

Виводиться перелік, потрібно заповнити комірку:

$$F_-: S_{b \rightarrow a, a}^{b, c} A2 = (a \rightarrow (b \rightarrow a)) \rightarrow ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a));$$

$$F_-: S_{b \rightarrow a}^b A1 = (a \rightarrow ((b \rightarrow a) \rightarrow a));$$

$$F_-: MP(F3, F2) = (a \rightarrow a);$$

$$F_-: MP(A1, F1) = ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a)).$$

Якщо відповідь правильна, тобто встановлено правильну послідовність, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! Перший крок формального доведення такий:

$$F1: S_{b \rightarrow a, a}^{b, c} A2 = (a \rightarrow (b \rightarrow a)) \rightarrow ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a))».$$

Крок 2. Користувачу виводиться завдання: «Довести, що $\neg(a \rightarrow a)$. Встановіть послідовність формального доведення. Вкажіть другий крок». Виводиться перелік, потрібно заповнити комірку:

$$F1: S_{b \rightarrow a, a}^{b, c} A2 = (a \rightarrow (b \rightarrow a)) \rightarrow ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a));$$

$$F_-: S_{b \rightarrow a}^b A1 = (a \rightarrow ((b \rightarrow a) \rightarrow a));$$

$$F_-: MP(F3, F2) = (a \rightarrow a);$$

$$F_-: MP(A1, F1) = ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a)).$$

Якщо відповідь правильна, тобто встановлено правильну послідовність, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! Другий крок формального доведення такий:

$$F2: MP(A1, F1) = ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a))».$$

Крок 3. Користувачу виводиться завдання: «Довести, що $\neg(a \rightarrow a)$. Встановіть послідовність формального доведення. Вкажіть останні кроки».

Виводиться перелік, потрібно заповнити комірки:

$$F1: S_{b \rightarrow a, a}^{b, c} A2 = (a \rightarrow (b \rightarrow a)) \rightarrow ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a));$$

$$F_-: S_{b \rightarrow a}^b A1 = (a \rightarrow ((b \rightarrow a) \rightarrow a));$$

$$F_-: MP(F3, F2) = (a \rightarrow a);$$

$$F2: MP(A1, F1) = ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a)).$$

Якщо відповідь правильна, тобто встановлено правильну послідовність, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! Останні кроки формального доведення такі:

$$F3: S_{b \rightarrow a}^b A1 = (a \rightarrow ((b \rightarrow a) \rightarrow a));$$

$$F4: MP(F3, F2) = (a \rightarrow a) ».$$

Крок 4. Користувачу виводиться завдання: «Довести, що $a / -(b \rightarrow a)$. Встановіть послідовність формального доведення».

Виводиться перелік, потрібно заповнити комірки:

$$F_-: MP(F1, A1) = (b \rightarrow a);$$

$$F_-: a.$$

Якщо відповідь правильна, тобто встановлено правильну послідовність, то відбувається перехід на наступний крок, в іншому разі – відображається повідомлення про помилку: «Помилка! Формальна вивідність така:

$$F1: a;$$

$$F2: MP(F1, A1) = (b \rightarrow a) ».$$

Крок 5. Користувачу виводиться завдання: «Довести, що $a, b, a \rightarrow (b \rightarrow c) / -c$. Встановіть послідовність формального доведення.

Вкажіть перший крок». Виводиться перелік, потрібно заповнити комірку:

$$F_-: a \rightarrow (b \rightarrow c);$$

$$F_-: MP(F1, F3) = b \rightarrow c;$$

$$F_-: MP(F2, F4) = c;$$

$$F_-: a;$$

$$F_-: b.$$

Якщо відповідь правильна, тобто встановлено правильну послідовність, то відбувається перехід на крок 6, в іншому разі – відображається повідомлення про помилку: «Помилка! Перший крок формального доведення такий:

$$F1: a».$$

Крок 6. Користувачу виводиться завдання: «Довести, що $a, b, a \rightarrow (b \rightarrow c) / -c$. Встановіть послідовність формального доведення. Вкажіть другий крок». Виводиться перелік, потрібно заповнити комірку:

$$F_-: a \rightarrow (b \rightarrow c);$$

$$F_-: MP(F1, F3) = b \rightarrow c;$$

$$F_-: MP(F2, F4) = c;$$

$$F1: a;$$

$$F_-: b.$$

Якщо відповідь правильна, тобто встановлено правильну послідовність, то відбувається перехід на крок 7, в іншому разі – відображається повідомлення про помилку: «Помилка! Другий крок формального доведення такий:

$$F2: b».$$

Крок 7. Користувачу виводиться завдання: «Довести, що $a, b, a \rightarrow (b \rightarrow c) / -c$. Встановіть послідовність формального доведення. Вкажіть третій крок». Виводиться перелік, потрібно заповнити комірку:

$$F_-: a \rightarrow (b \rightarrow c);$$

$$F_-: MP(F1, F3) = b \rightarrow c;$$

$$F_-: MP(F2, F4) = c;$$

$F1: a;$

$F2: b.$

Якщо відповідь правильна, тобто встановлено правильну послідовність, то відбувається перехід на крок 8, в іншому разі – відображається повідомлення про помилку: «Помилка! Третій крок формального доведення такий:

$$F3: a \rightarrow (b \rightarrow c)».$$

Крок 8. Користувачу виводиться завдання: «Довести, що $a, b, a \rightarrow (b \rightarrow c) / \neg c$. Встановіть послідовність формального доведення. Вкажіть останні кроки». Виводиться перелік, потрібно заповнити комірки:

$$F3: a \rightarrow (b \rightarrow c);$$

$$F_-: MP(F1, F3) = b \rightarrow c;$$

$$F_-: MP(F2, F4) = c;$$

$F1: a;$

$F2: b.$

Якщо відповідь правильна, тобто встановлено правильну послідовність, то відбувається перехід на крок 9 до результатів, в іншому разі – відображається повідомлення про помилку: «Помилка! Останні кроки формального доведення такі:

$$F4: MP(F1, F3) = b \rightarrow c;$$

$$F5: MP(F2, F4) = c».$$

Крок 9. Користувачу виводиться повідомлення про завершення: «Вітаємо! Ви ознайомилися з темою «Формальні доведення та вивідність формул». Пропонується завершити роботу тренажера або почати проходження спочатку.

3.4. Розробка блок-схеми

Оскільки алгоритм роботи тренажера поділяється на два блоки, то на рисунках 3.1 - 3.10 зображено блок-схему блоку «Формули і підстановки».

На рисунках 3.11 - 3.16 зображено блок-схему блоку «Формальні доведення та вивідність формул»

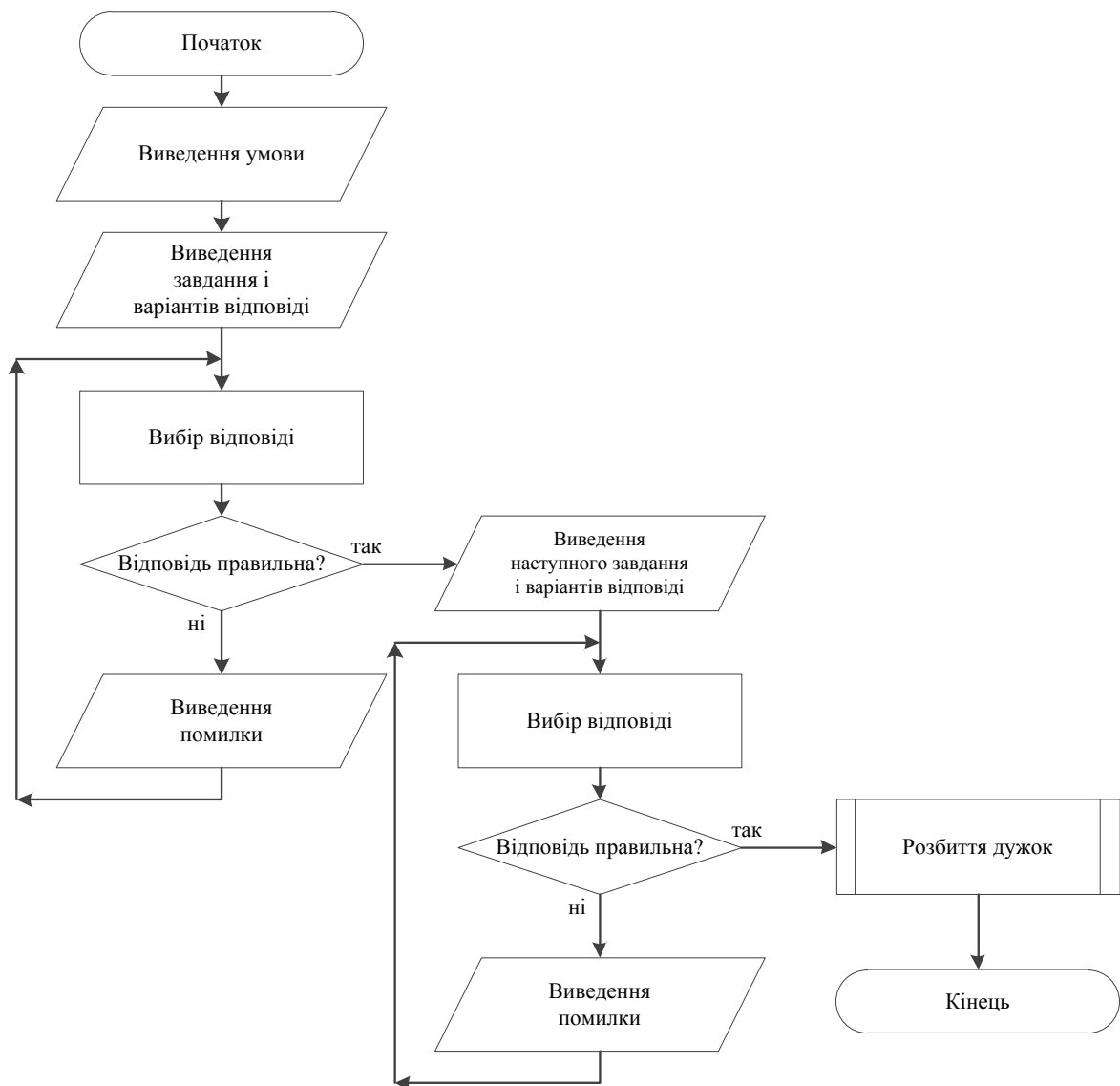


Рисунок 3.1 – Блок-схема блоку «Формули і підстановки»

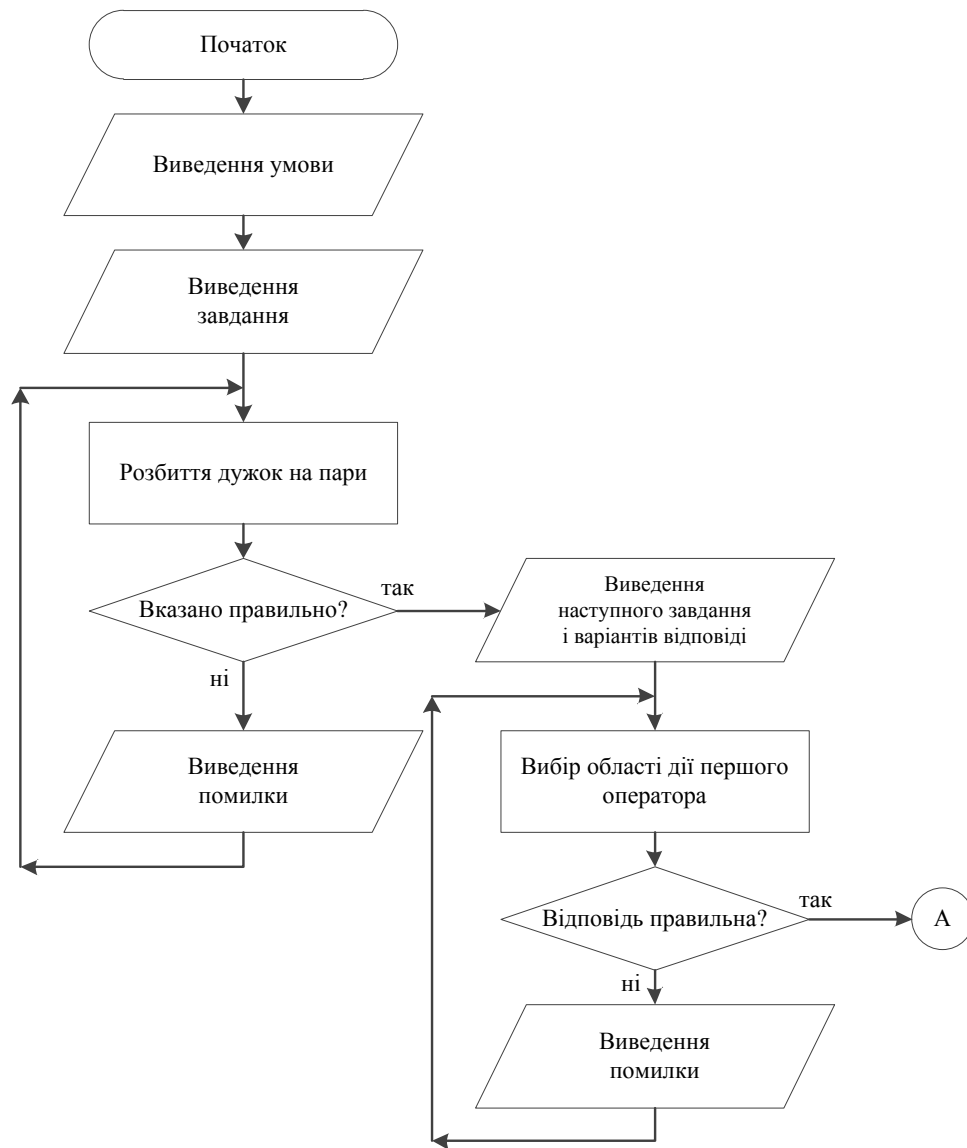


Рисунок 3.2 – Блок-схема підпроцесу «Розбиття дужок»

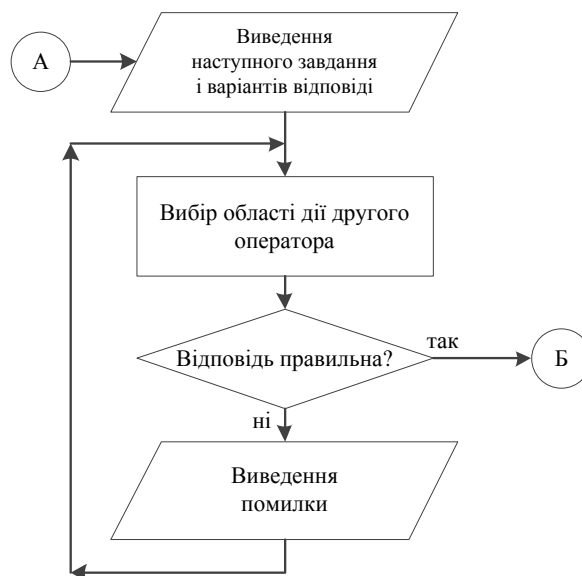


Рисунок 3.3 – Продовження блок-схеми підпроцесу «Розбиття дужок»

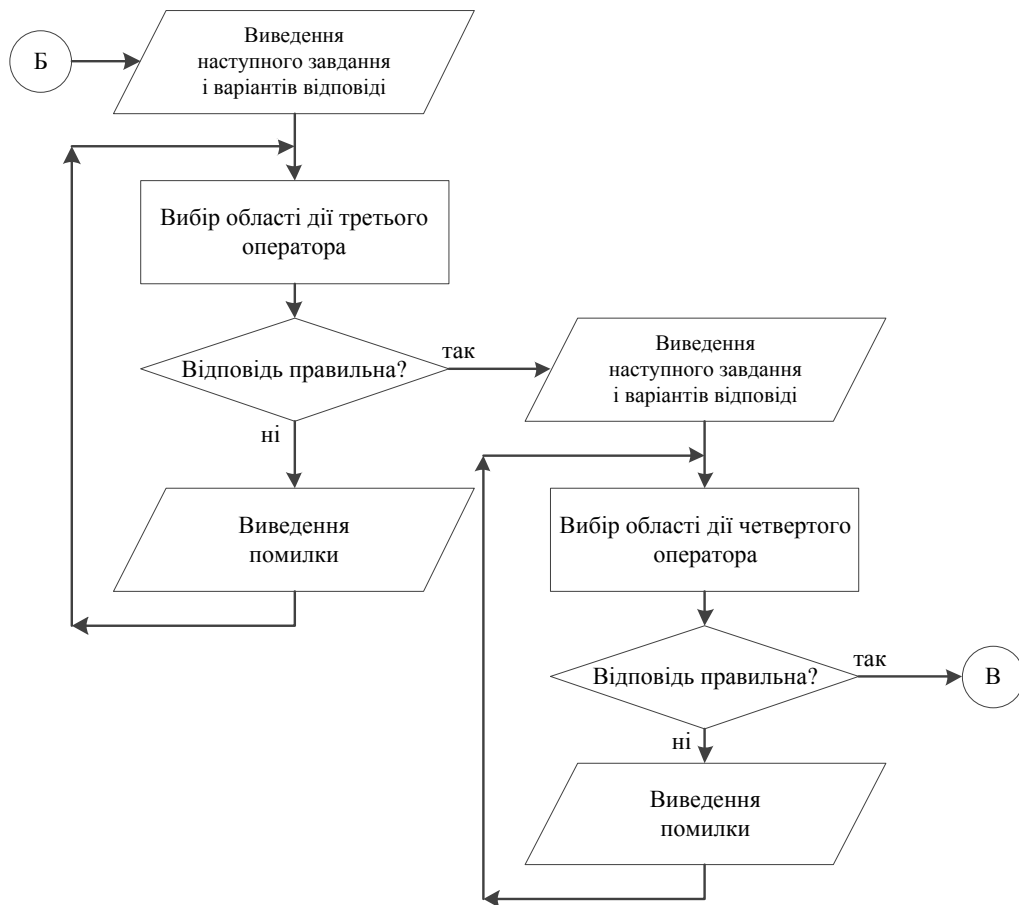


Рисунок 3.4 – Продовження блок-схеми підпроцесу «Розбиття дужок»

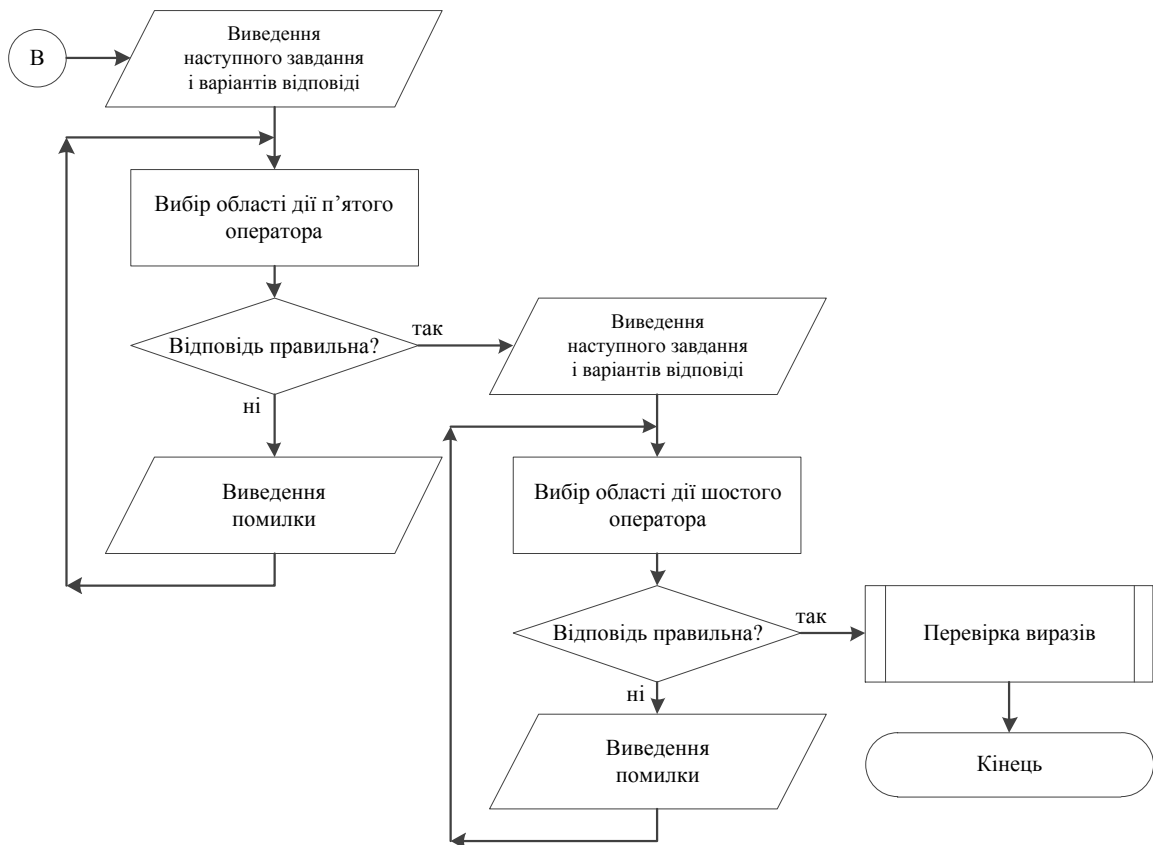


Рисунок 3.5 – Продовження блок-схеми підпроцесу «Розбиття дужок»

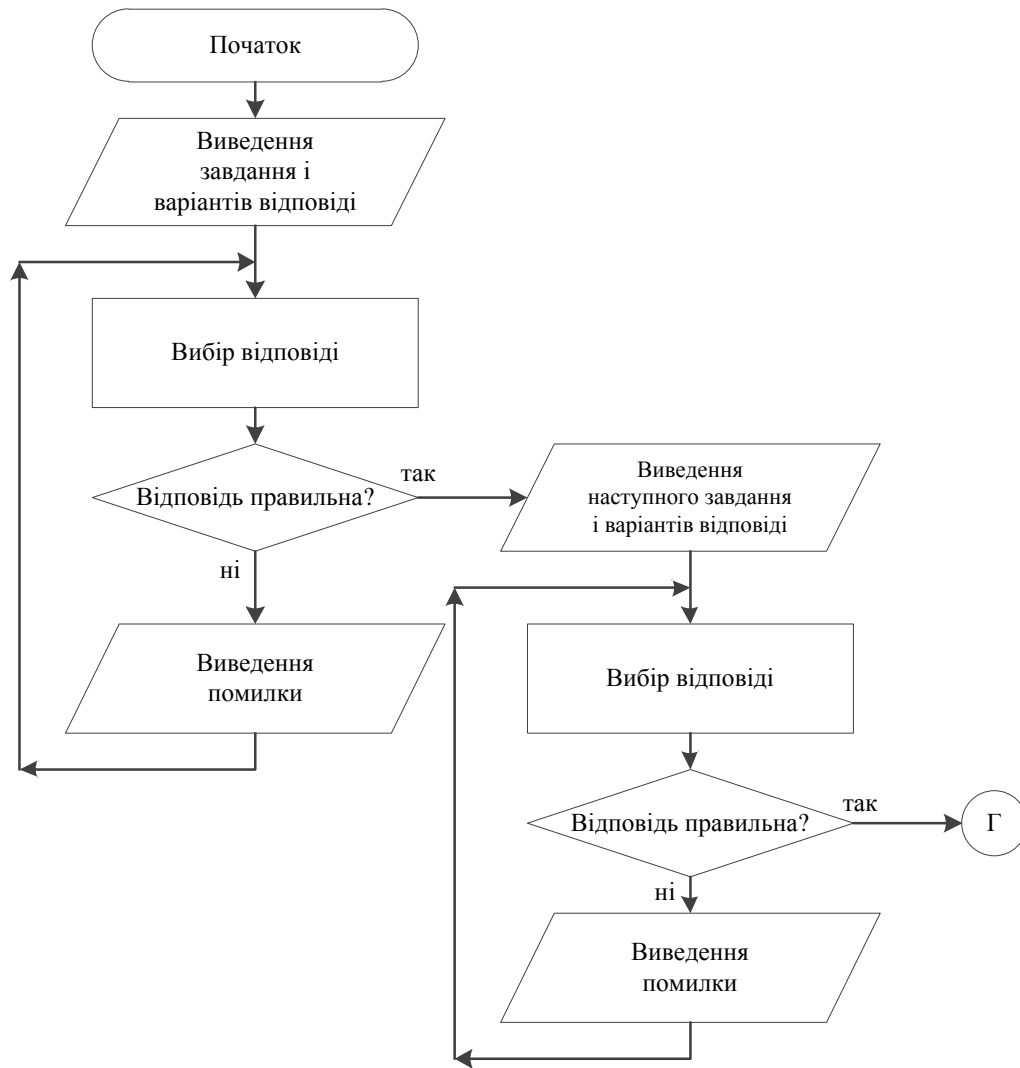


Рисунок 3.6 – Блок-схема підпроцесу «Перевірка виразів»

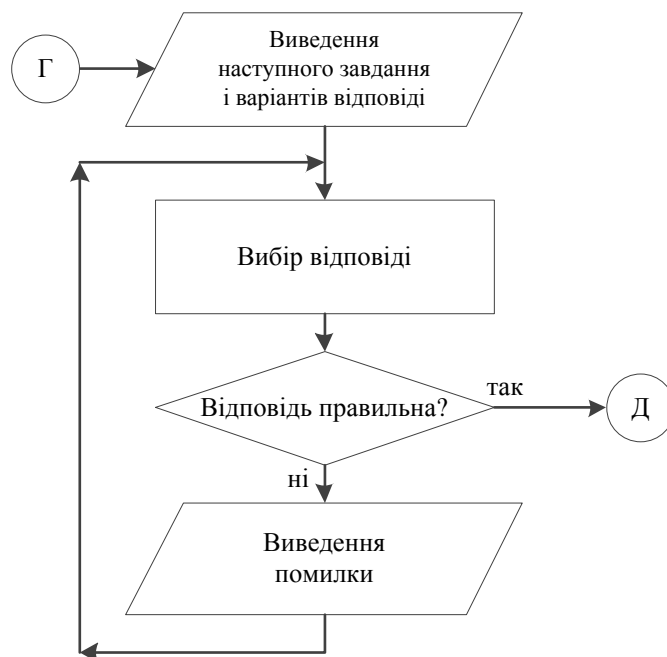


Рисунок 3.7 – Продовження блок-схеми підпроцесу «Перевірка виразів»

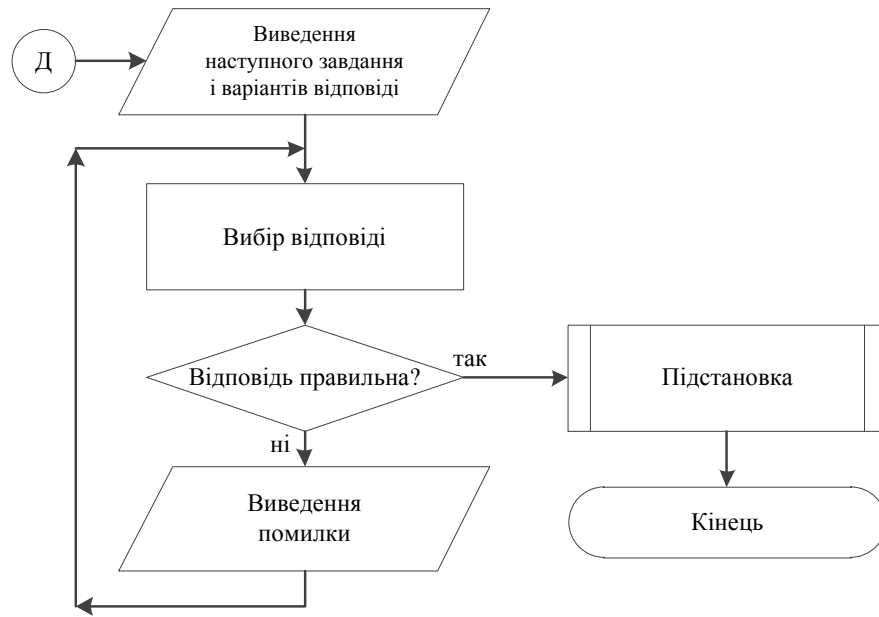


Рисунок 3.8 – Продовження блок-схеми підпроцесу «Перевірка виразів»

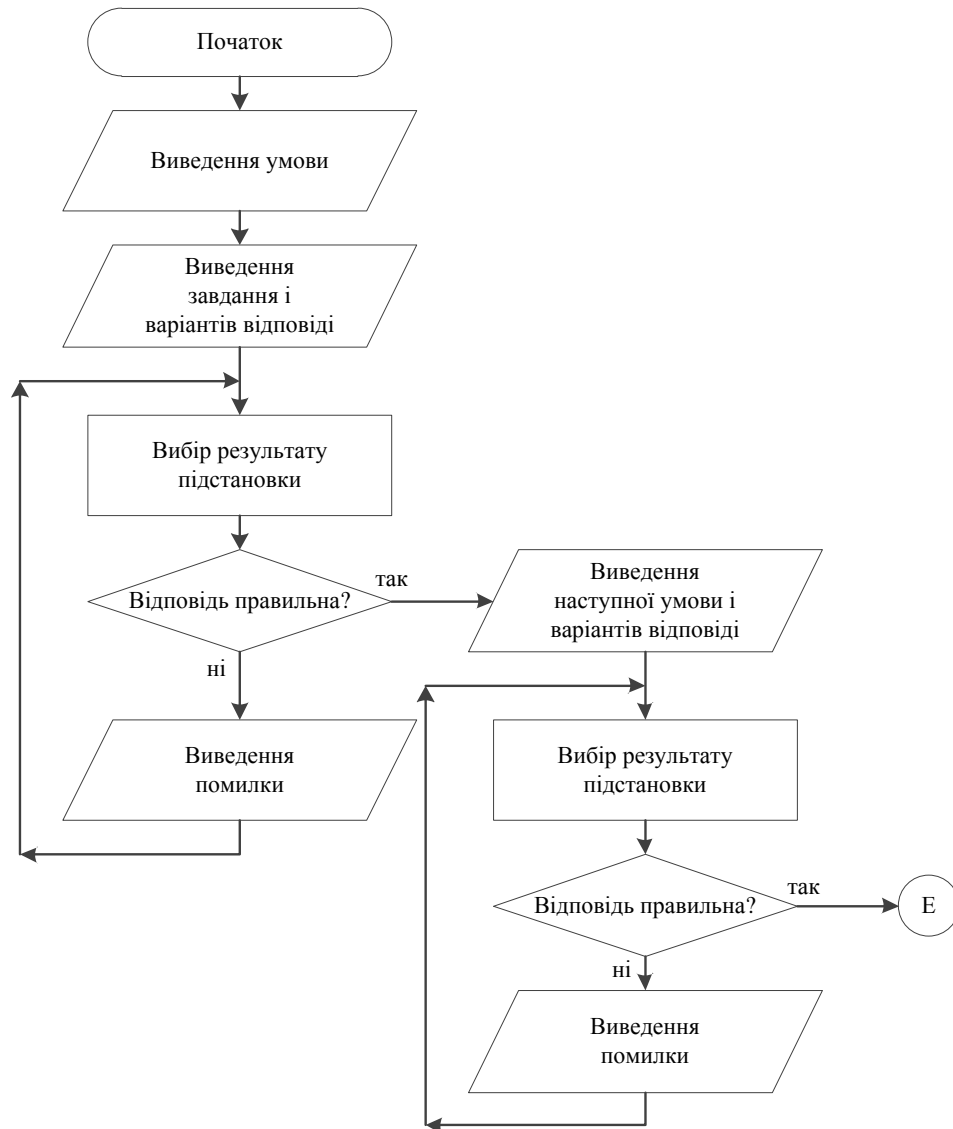


Рисунок 3.9 – Блок-схема підпроцесу «Підстановка»

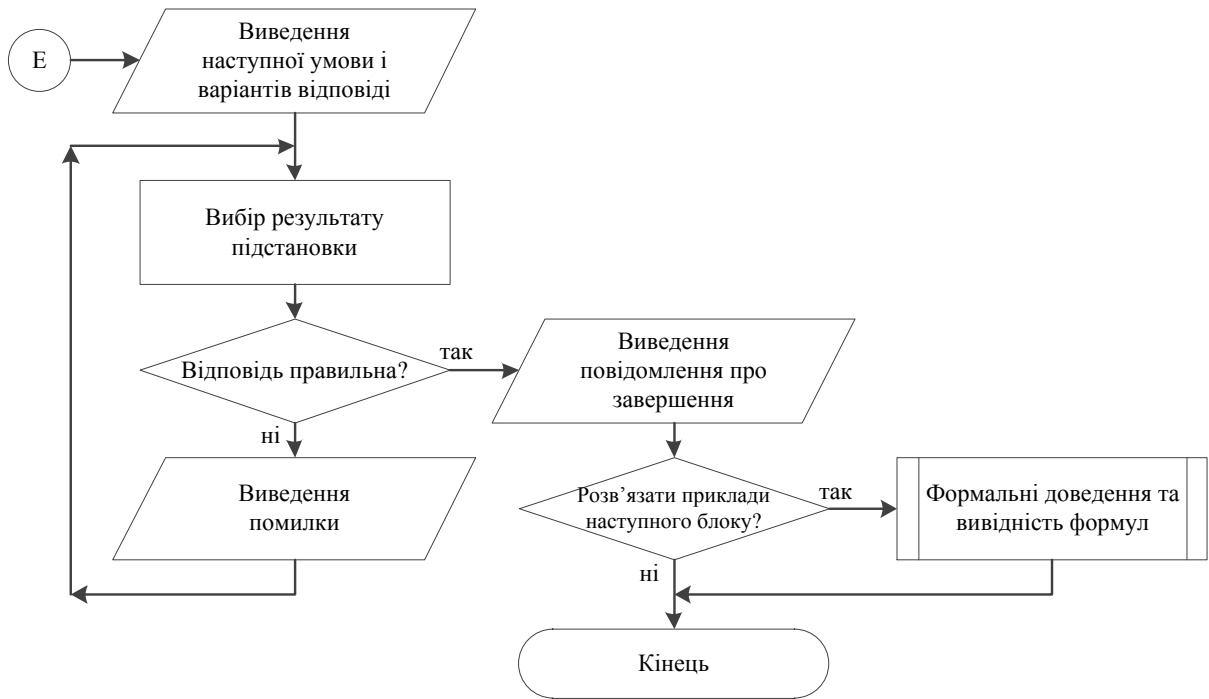


Рисунок 3.10 – Продовження блок-схеми підпроцесу «Підстановка»

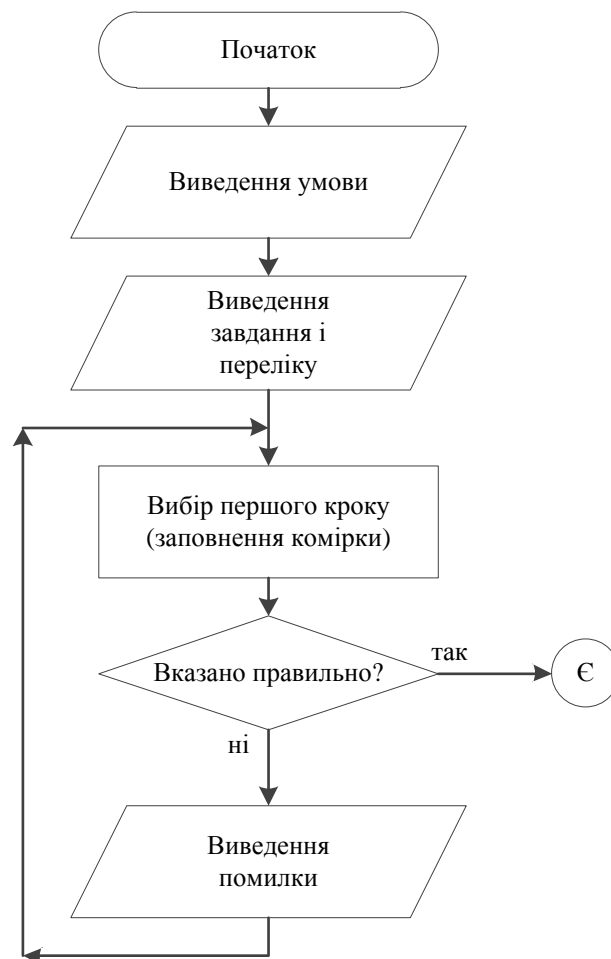


Рисунок 3.11 – Блок-схема блоку «Формальні доведення та вивідність формул»

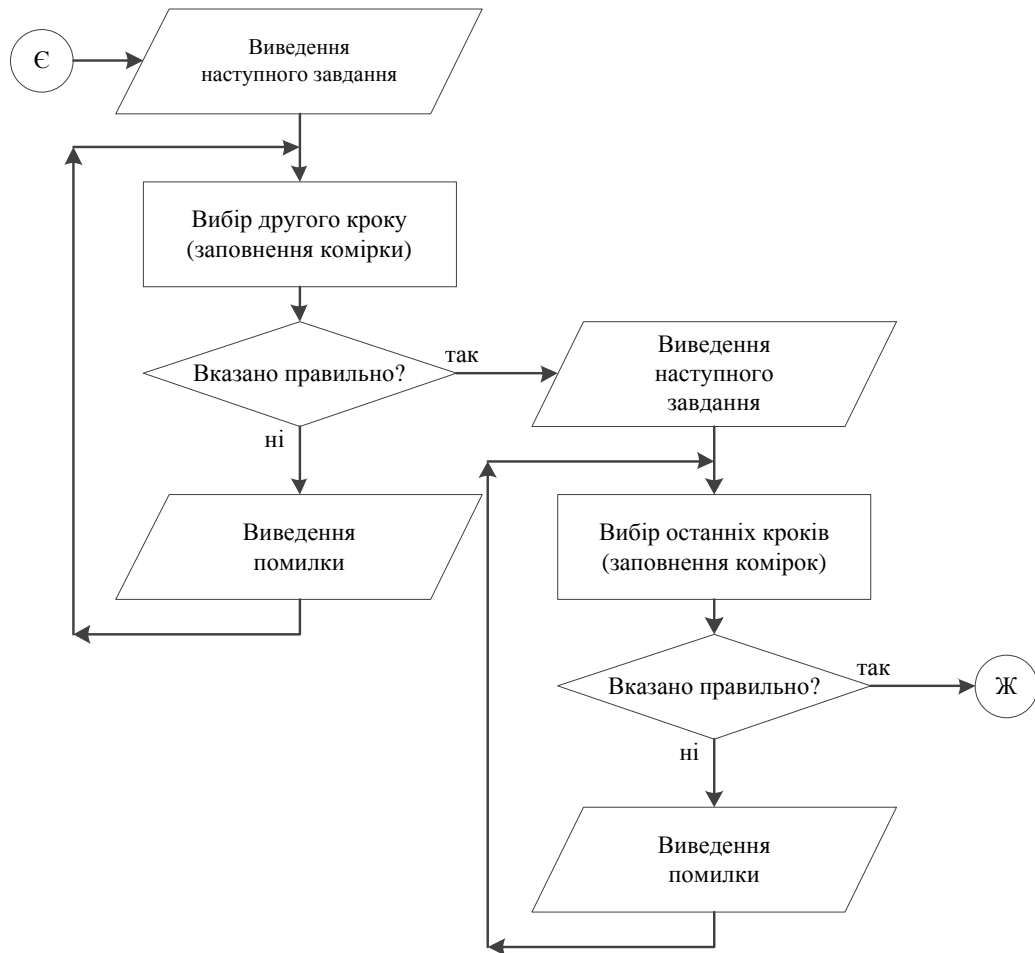


Рисунок 3.12 – Продовження блок-схеми блоку «Формальні доведення та вивідність формул»

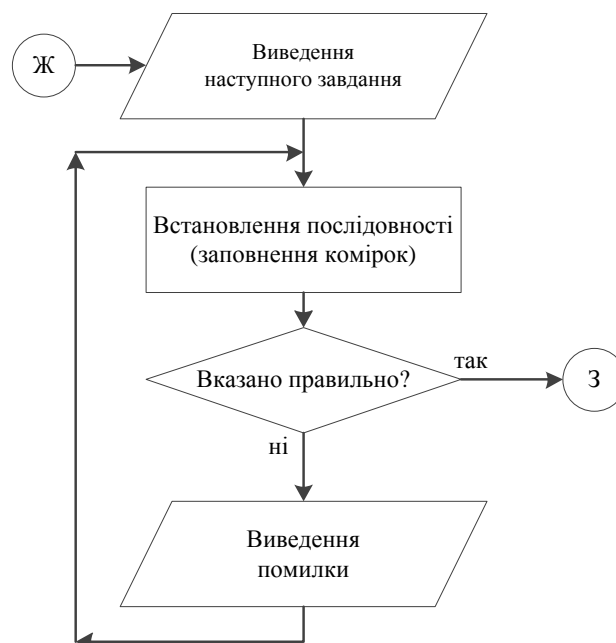


Рисунок 3.13 – Продовження блок-схеми блоку «Формальні доведення та вивідність формул»

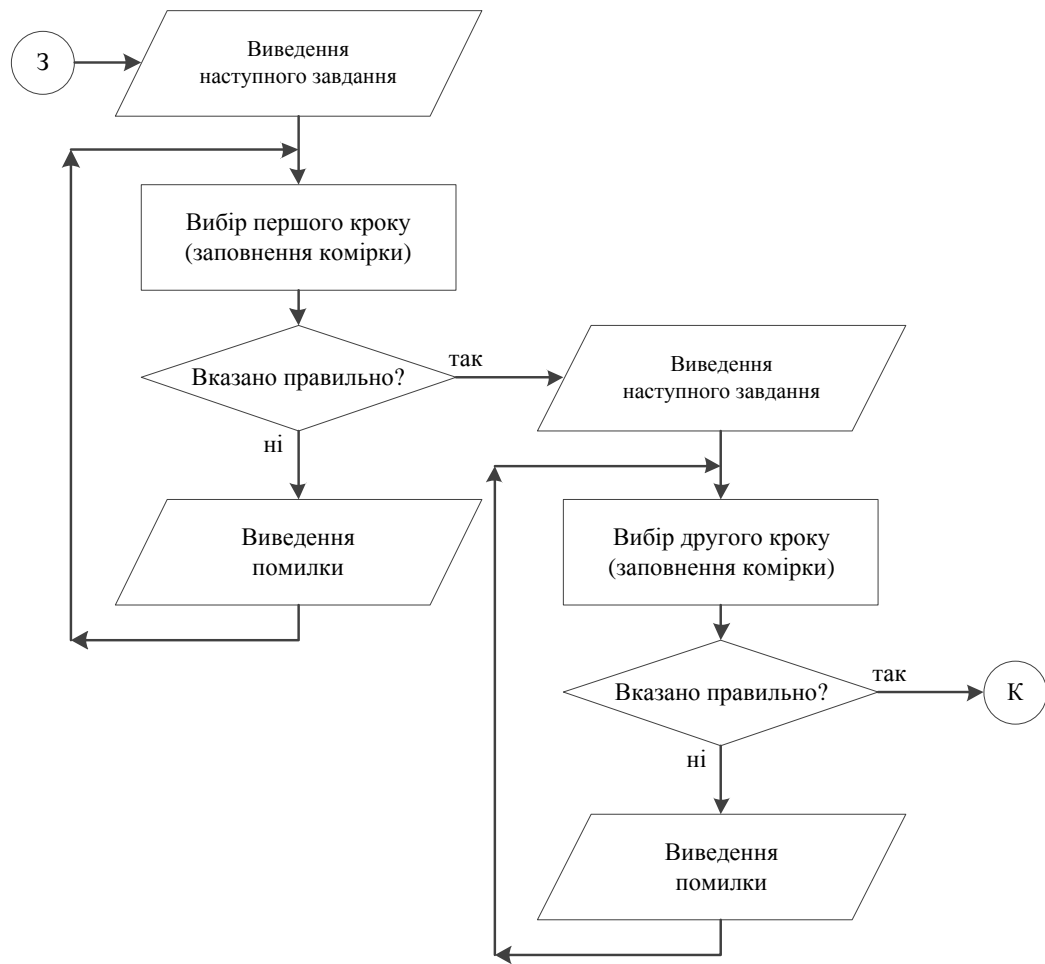


Рисунок 3.14 – Продовження блок-схеми блоку «Формальні доведення та вивідність формул»



Рисунок 3.15 – Продовження блок-схеми блоку «Формальні доведення та вивідність формул»

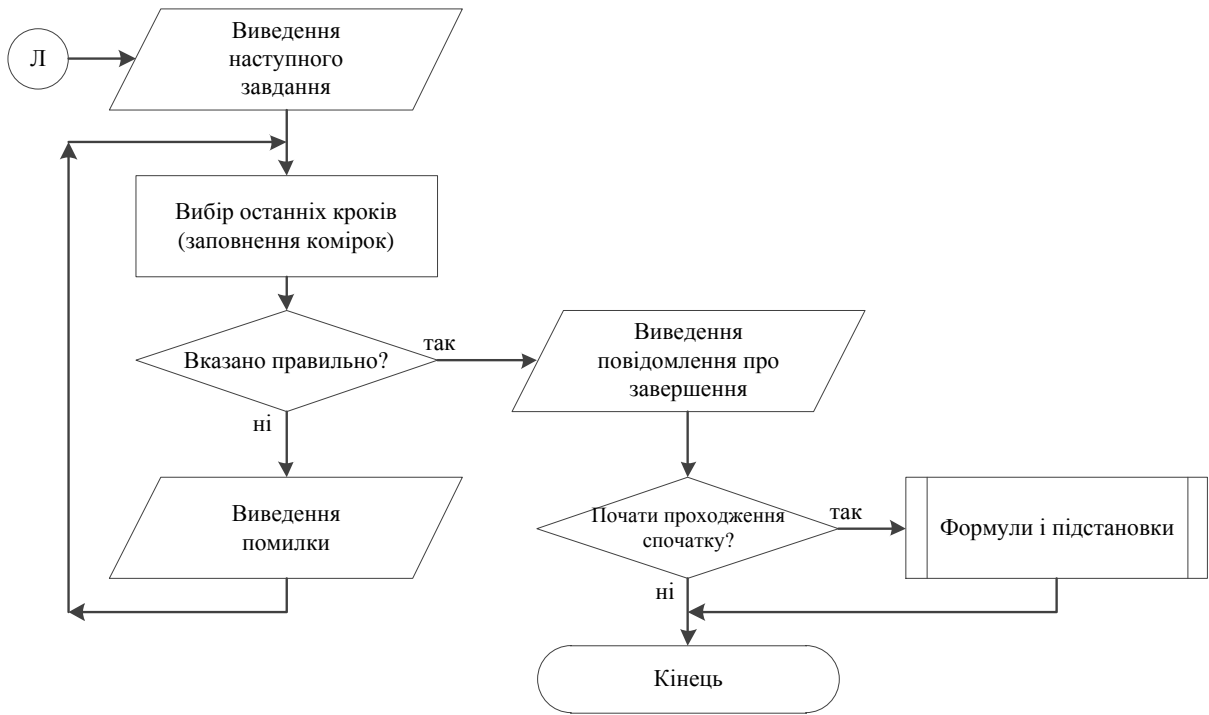


Рисунок 3.16 – Продовження блок-схеми блоку «Формальні доведення та вивідність формул»

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

Java є мовою програмування, за допомогою якої розробники програмного забезпечення (програмісти) створюють різні прикладні додатки для комп'ютерів, смартфонів, планшетів та інших інтелектуальних пристроїв. Особливістю програм на Java є те, що вони можуть запускатись на будь-яких комп'ютеризованих пристроях, які працюють під різними операційними системами, причому без повторної компіляції коду.

Для їх виконання необхідно лише встановити середовище для виконання – JRE (Java Runtime Environment), завантаживши його з сайту <http://www.oracle.com>. JRE розроблені для багатьох операційних систем – Linux(x86,x64), Mac OS X64, Solaris, Windows (x86,x64), завдяки чому код Java працює майже на всіх різновидах комп'ютерів та операційних систем.

JRE забезпечує безпечну та зручну роботу додатків на Java, тому користувачі можуть не турбуватись про несанкціоноване втручання до ресурсів свого персонального комп'ютера з боку стороннього Java коду. Необхідно лише періодично оновлювати JRE, на даний момент остання версія JRE 8 Update 171. Вбудована технологія забезпечення безпеки Java включає в себе значний набір API (Application Programming Interface) механізмів та додаткових інструментів, включаючи широковідомі та надійні алгоритми та протоколи безпеки.

Це передбачає використання криптографічних механізмів захисту інформації, інфраструктуру відкритих ключів, захищений зв'язок, автентифікацію та контроль доступу [10].

Нині близько 3 млрд мобільних телефонів працюють на Java, а також 125 млн телевізорів і кожен Blu-Ray програвач. Ця мова постійно посідає перші місця в рейтингах розробників програмного забезпечення як кращий вибір серед мов програмування. Крім цього, у Java величезна кількість

прихильників, які живуть за принципом WORA (Write once, run anywhere), що перекладається як “написав один раз, запускай скрізь”.

Java – дуже популярна мова програмування. З моменту запуску в 1995 році вийшло вже дев’ять основних версій. Java стала технічним феноменом, що багато в чому пов’язано з її унікальною портативністю: Java- додатки працюють на будь-якому пристрої або в операційній системі. Її основні якості – незалежність від архітектури, просте, але надійне виконання програм і гарантія безпеки.

Вісім фактів про Java, що охоплюють історію, популярність, останні модифікації мови та її талісман:

- ✓ Приблизно на 9 з 10 комп’ютерах у США встановлена віртуальна Java-машина, і на 97 % корпоративних комп’ютерів використовується ця мова.

- ✓ Спершу Java створювався для інтерактивного ТБ, але індустрія кабельного телебачення була недостатньо розвинена для того, щоб використати її.

- ✓ Мова розроблялася Джеймсом Гослінгом (James Gosling) і спочатку була названа Oak на честь дуба, який ріс біля офісу Гослінга. Згодом мову перейменували на Green, а пізніше вона набула назви Java, і використовувалася для написання клієнтських додатків і серверного програмного забезпечення. Своїй останній назві мова завдячує торговій марці кави Java, тому на офіційній емблемі зображена філіжанка з кавою, що парує. Відома й інша версія походження назви, пов’язана з алюзією на каву-машину як приклад побутового пристрою, для програмування якого спочатку створювалася мова. Зараз розвиток Java контролює Oracle, яка купила компанію Sun 2010 року.

- ✓ Середня зарплата Java-розробника \$83 975. По всьому світу налічується близько 9 млн Java- програмістів.

- ✓ Згідно з даними сайту Tiobe.com, Java посідає друге місце серед найпопулярніших мов програмування. Перше місце віддане мові С.

✓ Головну технологію Java -бібліотеку для модульного тестування ПЗ JUnit – використовують 4 з 5 програмістів. На другому місці Jenkins – його використовують 70 % розробників.

✓ Java Platform Standard Edition 8 – останній і головний реліз Java. Його особливості – підвищена ефективність розробки і продуктивність додатків завдяки зменшеній кількості стандартних вставок, поліпшеним колекціям і приміткам.

✓ Дюк (Duke), символ Java, був створений Джо Пелренгом (Joe Palrang), який згодом працював над популярними анімаційними фільмами, такими як Шрек [11].

Крім додатків, мова Java дозволяє створювати аплети (applets). Це програми, що працюють в середовищі іншої програми - браузера. Аплету не потрібне вікно верхнього рівня - їм служить вікно браузера. Вони не запускаються JVM — їх завантажує браузер, котрий сам запускає JVM для виконання аплету. Ці особливості відбиваються на написанні програми аплета.

З точки зору мови Java, аplet — це всяке розширення класу Applet, котрий, в свою чергу, розширяє клас Panel. Таким чином, аplet - це панель спеціального виду, контейнер для розміщення компонентів з додатковими властивостями і методами. Менеджером розміщення компонентів по замовчуванню, як і в класі Panel, служить FlowLayout. Клас Applet знаходиться в пакеті java.applet, в якому крім нього є тільки три інтерфейси, реалізовані в браузері. Треба відмітити, що не всі браузери реалізують ці інтерфейси повністю.

Оскільки JVM не запускає аplet, відпадає необхідність в методі main(), його немає в аплетах. В аплетах рідко зустрічається конструктор. Справа в тому, що при запуску першого створюється його контекст. Під час виконання конструктора контекст ще не сформований, тому не всі початкові значення вдається визначити в конструкторі. Початкові дії, зазвичай виконувані в конструкторі і методі main(), в аплеті записуються в

метод `init()` класу `Applet`. Цей метод автоматично запускається виконуючою системою Java браузера зразу ж після завантаження аплету [12].

NetBeans IDE — вільне інтегроване середовище розробки (IDE) для мов програмування Java, JavaFX, C/C++, PHP, JavaScript, HTML5, Python, Groovy. Середовище може бути встановлене і для підтримки окремих мов, і у повній конфігурації. Середовище розробки NetBeans за замовчуванням підтримує розробку для платформ J2SE і J2EE.

Поширюється у сирцевих текстах під ліцензією Apache License. Проект NetBeans IDE підтримувався і спонсорувався фірмою Sun Microsystems і після придбання Sun — Oracle. У жовтні 2016 року Oracle передав NetBeans у власність Apache Software Foundation, яка займається розробкою і підтримкою проекту.

NetBeans IDE доступна для платформ Microsoft Windows, GNU/Linux, FreeBSD, і Solaris (як SPARC, так x86). Для інших платформ доступна можливість зібрати NetBeans самостійно із сирцевих текстів.

За якістю і можливостям останні версії NetBeans IDE змагається з найкращими інтегрованими середовищами розробки для мови Java, підтримуючи рефакторинг, профілювання, виділення синтаксичних конструкцій кольором, автодоповнення мовних конструкцій на льоту, шаблони коду та інше.

NetBeans IDE підтримує плагіни, дозволяючи розробникам розширювати можливості середовища [13].

4.2. Опис процесу програмної реалізації

Першим кроком було розроблено графічне представлення (рис. 4.1). Так на панелі Номе розміщено основну інформацію про тренажер:

- тема тренажера;
- виконавець роботи;

- науковий керівник;
- кнопка переходу до теоретичного матеріалу;
- кнопка початку проходження тренажера;
- селектор для вибору мови.

Панель Examples містить два блоки прикладів, поділених відповідно до кроків алгоритму. На панелі End виводиться повідомлення про завершення тренажера, надається можливість перейти до наступного чи попереднього блоку прикладів, завершити роботу.

Help містить вкладки з теоретичним матеріалом і прикладами застосування.

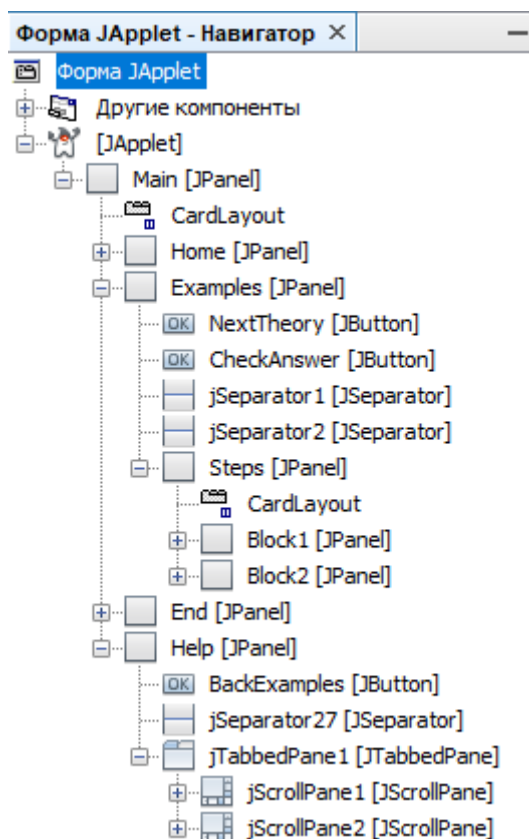


Рисунок 4.1 – Графічне представлення тренажера

Другим кроком підключено бібліотеки для подальшого використання елементів CardLayout, List, ArrayList, ImageIcon, JLabel.

```
import java.awt.CardLayout;
import java.util.List;
```

```
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
```

Ініціалізація і створення аплету відбувається за допомогою наступного коду. При цьому всі помилки додаються в список errors і для них всіх встановлюється властивість «не відображати» (setVisible(false)).

```
@Override
public void init() {
    /* Create and display the applet */
    try {
        java.awt.EventQueue.invokeLaterAndWait(new Runnable() {
            public void run() {
                initComponents();
                errors.add(Error1);
                errors.add(Error2);
                errors.add(Error3);
                errors.add(Error4);
                errors.add(Error5);
                errors.add(Error6);
                errors.add(Error7);
                errors.add(Error8);
                errors.add(Error9);
                errors.add(Error10);
                errors.add(Error11);
                errors.add(Error12);
                errors.add(Error13);
                errors.add(Error14);
                errors.add(Error15);
                errors.add(Error16);
```

```

        for(JLabel error : errors) {
            error.setVisible(false);
        }
    }
});
} catch (Exception ex) {
    ex.printStackTrace();
}
}

```

Оголошення змінних:

```

CardLayout card;
int b1step = 1;
int b2step = 1;
boolean home = true;
boolean block1 = true;
boolean block2 = false;
List<JLabel> errors = new ArrayList<>();

```

Наступним кроком було створено функції, в яких винесено важливі аспекти програми. Так функція `showBlock(int i)` відображає панель з поточним кроком відповідного блоку прикладів. Номер блоку вказується через параметр `int i`:

- 1 – «Формули і підстановки»;
- 2 – «Формальні доведення та вивідність формул».

```

private void showBlock(int i) {
    card = (CardLayout) Main.getLayout();
    card.show(Main, "examples");
    card = (CardLayout) Steps.getLayout();
    if(i == 1) {

```



```

        card.show(Steps, "block1");
    } else {
        card.show(Steps, "block2");
    }
}

```

Для виведення необхідного кроку у першому блоці прикладів розроблено `block1ShowStep(int i)`, де `int i` – номер кроку. Якщо вказати останній крок, то відобразиться панель з повідомленням про ознайомлення з даною темою.

```

private void block1ShowStep(int i) {
    if(i == 17) {
        card = (CardLayout) Main.getLayout();
        card.show(Main, "end");
    } else {
        card = (CardLayout) Block1.getLayout();
        card.show(Block1, "step"+i);
    }
}

```

Для другого блоку прикладів аналогічно – `block2ShowStep(int i)`.

```

private void block2ShowStep(int i) {
    if(i == 9) {
        card = (CardLayout) Main.getLayout();
        card.show(Main, "end");
    } else {
        card = (CardLayout) B2Steps.getLayout();
        card.show(B2Steps, "b2step"+i);
    }
}

```

Процес виведення повідомлення про помилку реалізовано по різному для кожного блоку прикладів. Так для першого просто змінюється властивість на «показувати»(`setVisible(true)`), а для другого – відбувається перехід між панелями. Тому саме для цього було створено функцію `block2ShowError(int i)`, де `int i` – номер кроку.

```
private void block2ShowError(int i) {
    card = (CardLayout) B2Steps.getLayout();
    card.show(B2Steps, "b2step"+i+"err");
    if(Language.getSelectedIndex() == 0) {
        CheckAnswer.setText("Продовжити");
    } else {
        CheckAnswer.setText("Continue");
    }
}
```

Оскільки тренажер має можливість переключатися між українською і англійською мовами, то розроблено функцію `setLanguage()` для зміни всього вмісту тренажера на вибрану мову (див. Додаток А).

Щоб всі кнопки функціонували потрібним чином на кожну з них створено відповідну подію. Подія `StartExamplesActionPerformed` спрацьовує при натисненні «Розпочати» і виводить завдання першого кроку алгоритму.

```
private void StartExamplesActionPerformed(java.awt.event.ActionEvent
evt) {
    home = false;
    setLanguage();
    showBlock(1);
    block1ShowStep(1);
}
```

При натисненні «Переглянути теорію» відбувається подія `StartTheoryActionPerformed`, при якій відбувається перехід до теоретичного матеріалу і прикладів.

```
private void StartTheoryActionPerformed(java.awt.event.ActionEvent
evt) {
    card = (CardLayout) Main.getLayout();
    card.show(Main, "help");
}
```

Аналогічно для «Перейти до теорії».

```
private void NextTheoryActionPerformed(java.awt.event.ActionEvent
evt) {
    card = (CardLayout) Main.getLayout();
    card.show(Main, "help");
}
```

Щоб повернутися назад використовується `BackExamplesActionPerformed`, спрацьовує при натисненні кнопки «Повернутися назад».

```
private void BackExamplesActionPerformed(java.awt.event.ActionEvent
evt) {
    if(home) {
        card = (CardLayout) Main.getLayout();
        card.show(Main, "home");
    } else {
        card = (CardLayout) Main.getLayout();
        card.show(Main, "examples");
    }
}
```

Подія `CheckAnswerActionPerformed` перевіряє надану відповідь, виконується якщо натиснути «Перевірити відповідь». При цьому якщо відповідь правильна – перехід до наступного кроку, якщо ні – виведення помилки (див. Додаток А).

При натисненні кнопки «Розв’язати приклади наступного блоку» очищуються результати проходження першого блоку прикладів і виводиться перше завдання другого блоку.

```
private void NextBlock2ActionPerformed(java.awt.event.ActionEvent
evt) {
    showBlock(2);
    block1 = false;
    block2 = true;
    b2step = 1;
    block2ShowStep(b2step);
    buttonGroup1.clearSelection();
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
    jTextField7.setText("");
    jTextField8.setText("");
    jTextField9.setText("");
    jTextField10.setText("");
    jTextField11.setText("");
    jTextField12.setText("");
    jLabel206.setEnabled(false);
    NextBlock2.setEnabled(false);
    Block1Exit.setEnabled(false);
}
```

```

jLabel207.setEnabled(true);
NextBlock1.setEnabled(true);
Block2Exit.setEnabled(true);
}

```

Якщо завершити другий блок та натиснути «Почати проходження спочатку» теж очищуються результати і відбувається перехід вже до першого – NextBlock1ActionPerformed (див. Додаток А).

Події Block1ExitActionPerformed і Block2ExitActionPerformed закривають тренажер. Вони відбуваються якщо натиснути «Завершити роботу тренажера».

```

private void Block1ExitActionPerformed(java.awt.event.ActionEvent
evt) {
    System.exit(0);
}
private void Block2ExitActionPerformed(java.awt.event.ActionEvent
evt) {
    System.exit(0);
}

```

Для переключення вмісту головної сторінки і сторінки з теоретичним матеріалом, прикладами в залежності від вибраної мови реалізовано LanguageItemStateChanged (повна версія – див. Додаток А).

```

private void LanguageItemStateChanged(java.awt.event.ItemEvent evt)
{
    if(Language.getSelectedIndex() == 0) {
        jLabel211.setText("Вибір мови:");
        jLabel208.setText("<html><center>Вітаємо в ...");
        jLabel209.setText("<html><b>Виконавець...");
        jLabel210.setText("<html><b>Науковий керівник...");
    }
}

```

```

StartTheory.setText("Переглянути теорію");
StartExamples.setText("Розпочати");

BackExamples.setText("< Повернутися назад >");
jTabbedPane1.setTitleAt(0, "Теорія");
jTabbedPane1.setTitleAt(1, "Приклади");
jLabel128.setIcon(new
ImageIcon(getClass().getResource("/images/theory-main.png")));
jLabel129.setIcon(new
ImageIcon(getClass().getResource("/images/theory-examples.png")));
} else {
jLabel211.setText("Language selection:");
jLabel208.setText("<html><center>Congratulations ...");
jLabel209.setText("<html><b>Contractor:</b> ...");
jLabel210.setText("<html><b>Supervisor:</b>...");
StartTheory.setText("Review the theory");
StartExamples.setText("Start");

BackExamples.setText("< Go back >");
jTabbedPane1.setTitleAt(0, "Theory");
jTabbedPane1.setTitleAt(1, "Examples");
jLabel128.setIcon(new
ImageIcon(getClass().getResource("/images/theory-main-eng.png")));
jLabel129.setIcon(new
ImageIcon(getClass().getResource("/images/theory-examples-eng.png")));
}
}

```

4.3. Опис програми

На головній сторінці виводиться тема тренажера, виконавець роботи і науковий керівник. Надається можливість вибрати українську (по замовчуванню) чи англійську мову. Пропонується розглянути теоретичний матеріал і приклади або розпочати проходження (рис. 4.2).

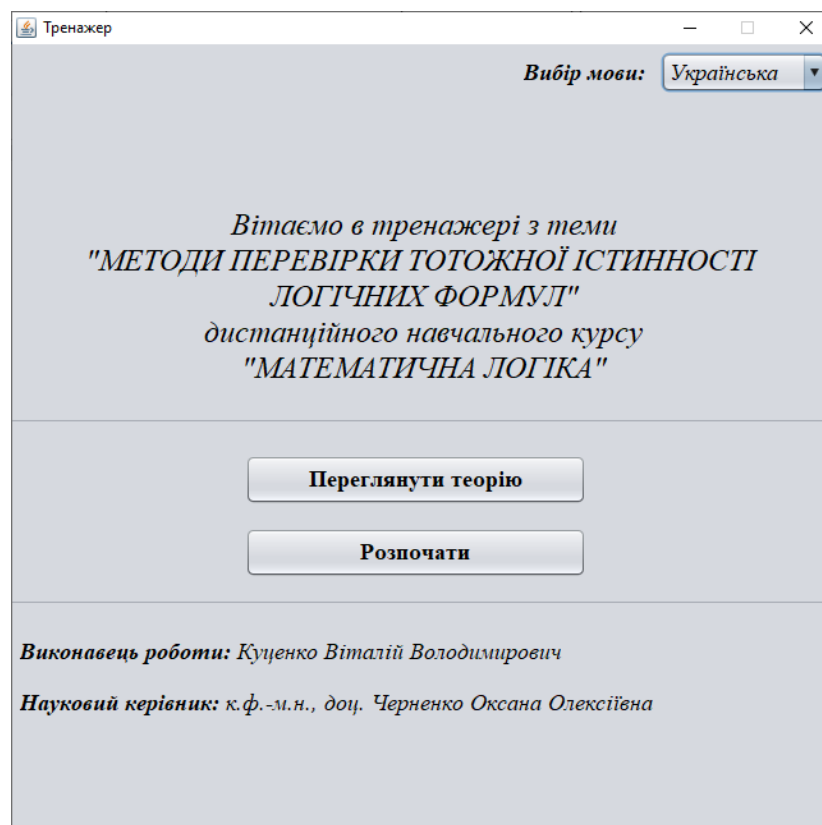


Рисунок 4.2 – Головна сторінка тренажера

Сторінка з теоретичним матеріалом має дві вкладки (рис. 4.3):

- Теорія (основна інформація);
- Приклади (приклад застосування).

Якщо вибрано англійську мову, то вміст головної сторінки і теорії буде виводитися цією мовою (рис. 4.4, 4.5).

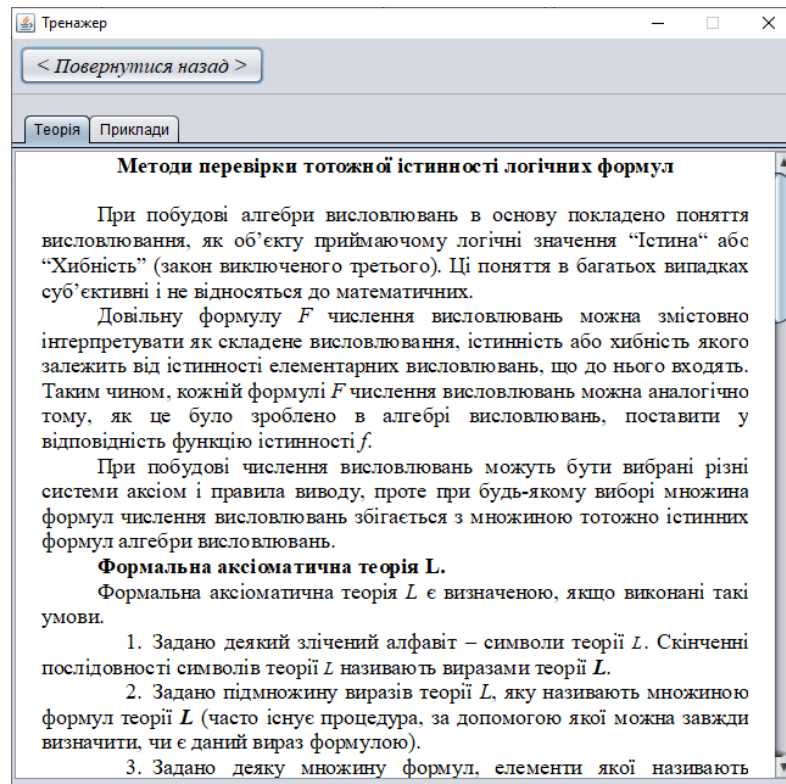


Рисунок 4.3 – Сторінка з теоретичним матеріалом

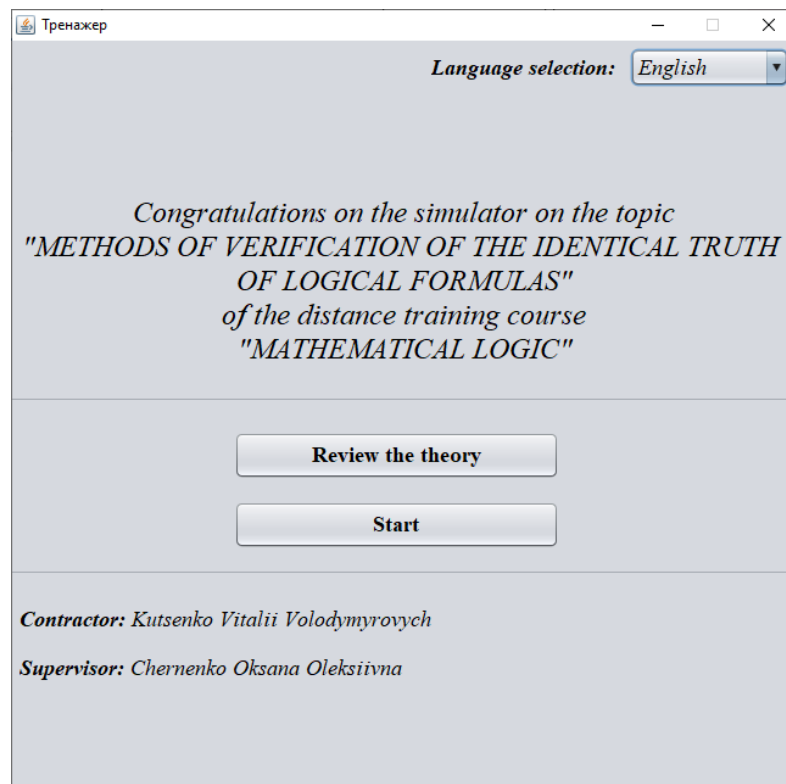


Рисунок 4.4 – Головна сторінка тренажера англійською мовою

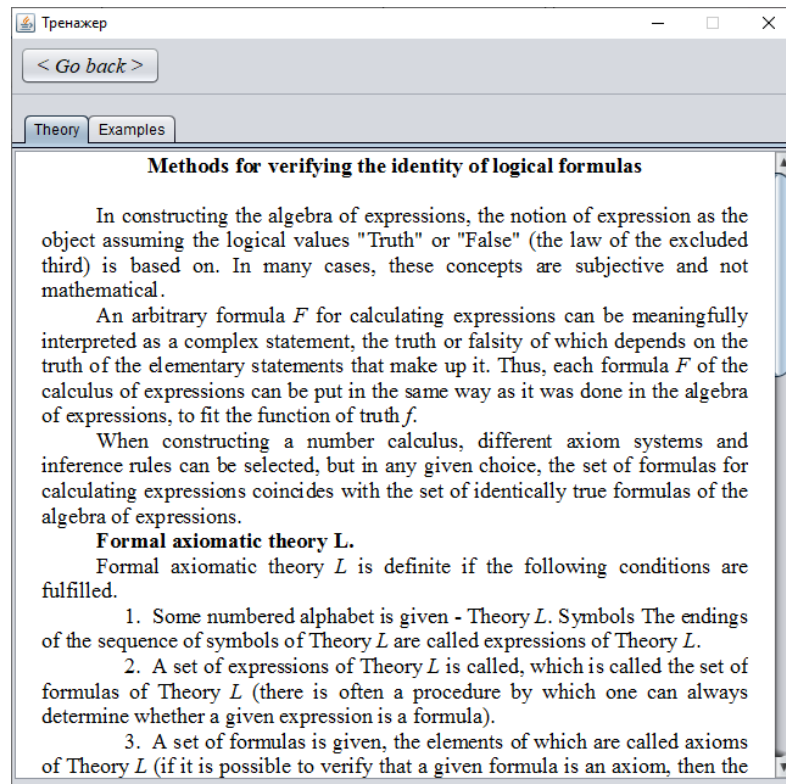


Рисунок 4.5 – Сторінка з теоретичним матеріалом англійською мовою

В тренажері реалізовано декілька типів завдань: вибір одного з варіантів відповідей, заповнення всіх активних полів, введення значення в одне із полів (рис. 4.6 – 4.8).

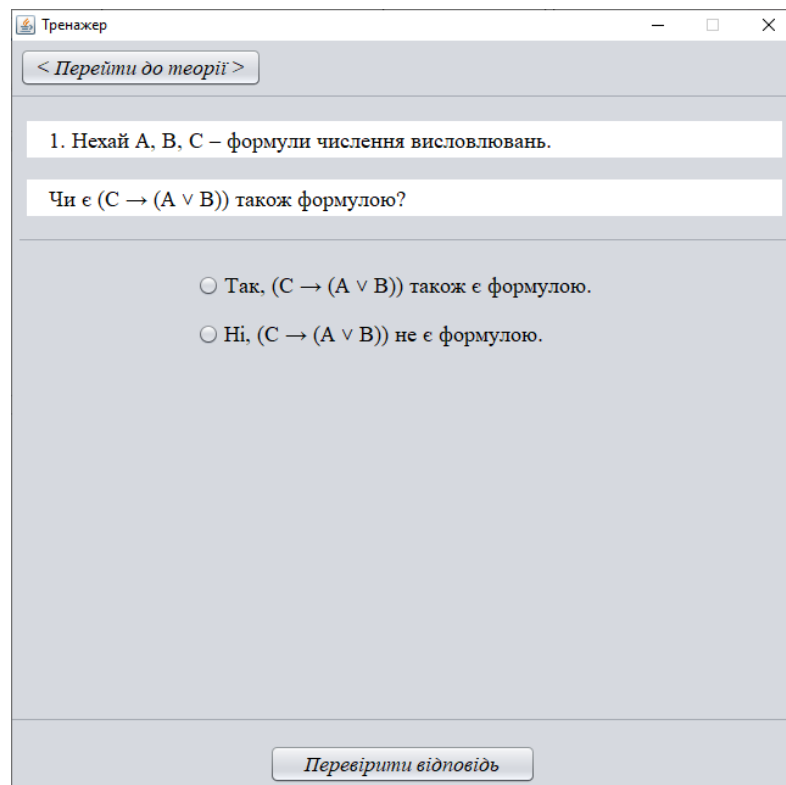


Рисунок 4.6 – Вибір одного з варіантів відповідей

Тренажер

< Перейти до теорії >

2. Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$.

Розбити дужки на пари.

$(\quad (\quad A \rightarrow B \quad) \rightarrow (\quad (\quad A \rightarrow C \quad) \rightarrow$
 $\rightarrow (\quad A \rightarrow (\quad B \wedge C \quad) \quad) \quad) \quad)$

Перевірити відповідь

Рисунок 4.7 – Заповнення всіх активних полів

Тренажер

< Перейти до теорії >

У якості аксіом числення висловлень візьмемо такі формули:

A1. $a \rightarrow (b \rightarrow a)$ A6. $a \rightarrow (a \vee b)$
 A2. $(a \rightarrow b) \rightarrow ((a \rightarrow (b \rightarrow c)) \rightarrow (a \rightarrow c))$ A7. $b \rightarrow (a \vee b)$
 A3. $(a \wedge b) \rightarrow a$ A8. $(a \rightarrow c) \rightarrow ((b \rightarrow c) \rightarrow ((a \vee b) \rightarrow c))$
 A4. $(a \wedge b) \rightarrow b$ A9. $(a \rightarrow \neg b) \rightarrow (b \rightarrow \neg a)$
 A5. $(a \rightarrow b) \rightarrow ((a \rightarrow c) \rightarrow (a \rightarrow (b \wedge c)))$ A10. $\neg \neg a \rightarrow a$

У якості правил виведення беремо правило підстановки і правило висновку (modus ponens, скорочено – MP).

Довести, що $\vdash (a \rightarrow a)$. Встановіть послідовність формального доведення. Вкажіть перший крок.

F ☐ : $S_{b \rightarrow a}^{b,c} A2 = (a \rightarrow (b \rightarrow a)) \rightarrow ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a));$
 F ☐ : $S_{b \rightarrow a}^b A1 = (a \rightarrow ((b \rightarrow a) \rightarrow a));$
 F ☐ : $MP(F3, F2) = (a \rightarrow a);$
 F ☐ : $MP(A1, F1) = ((a \rightarrow ((b \rightarrow a) \rightarrow a)) \rightarrow (a \rightarrow a)).$

Перевірити відповідь

Рисунок 4.8 – Введення значення в одне із полів

В тренажері реалізовано виведення повідомлення про помилку, якщо відповіді не правильно (рис. 4.9).

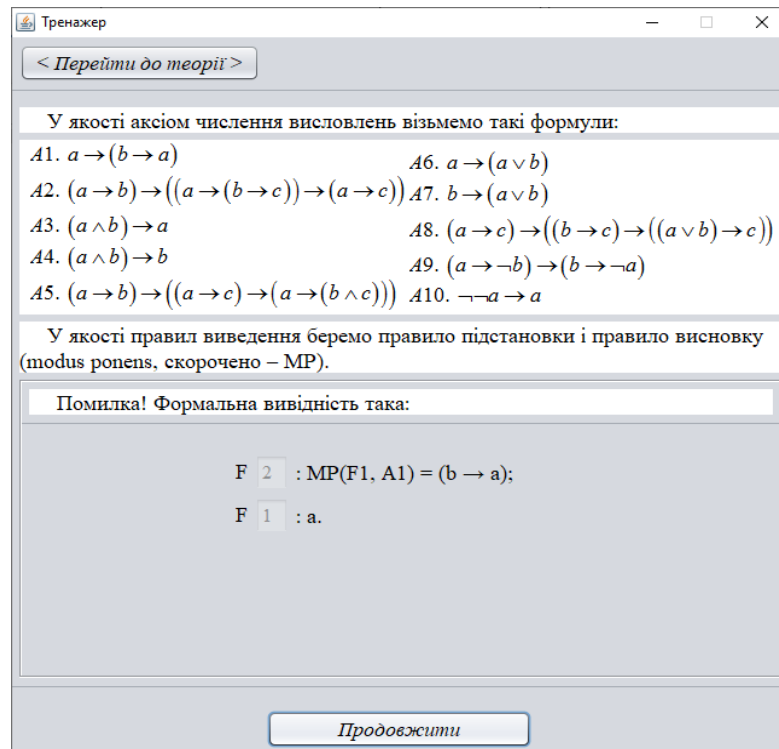


Рисунок 4.9 – Повідомлення про помилку

В кінці кожного блоку прикладів виводиться повідомлення про ознайомлення з темою (рис. 4.10).

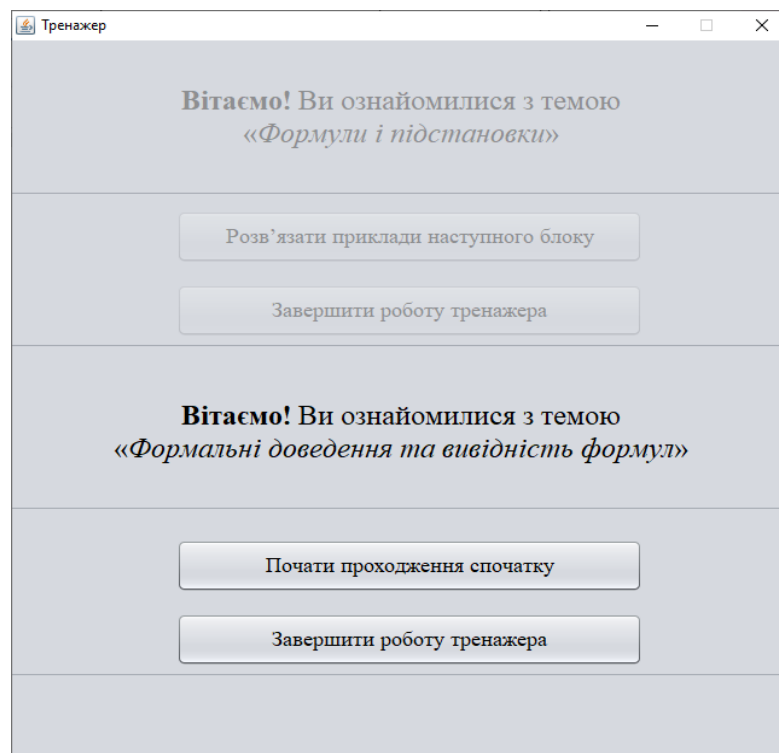


Рисунок 4.10 – Повідомлення про ознайомлення з темою

4.4. Інструкція по використанню тренажера

На головній сторінці першим чином слід вибрати мову, на якій буде виводиться вміст тренажера. Після початку проходження тренажера цієї можливості вже не буде (рис. 4.2).

Щоб переглянути теоретичний матеріал потрібно натиснути кнопку «Переглянути теорію» (рис. 4.3).

Для того щоб почати проходження необхідно натиснути кнопку «Розпочати». При цьому виведеться умова і завдання першого кроку блоку «Формули і підстановки» (рис. 4.5).

Тепер потрібно вибрати відповідь і натиснути «Перевірити відповідь». Якщо відповіді неправильно, то з'явиться повідомлення про помилку (рис. 4.11), інакше – перехід до наступного кроку.

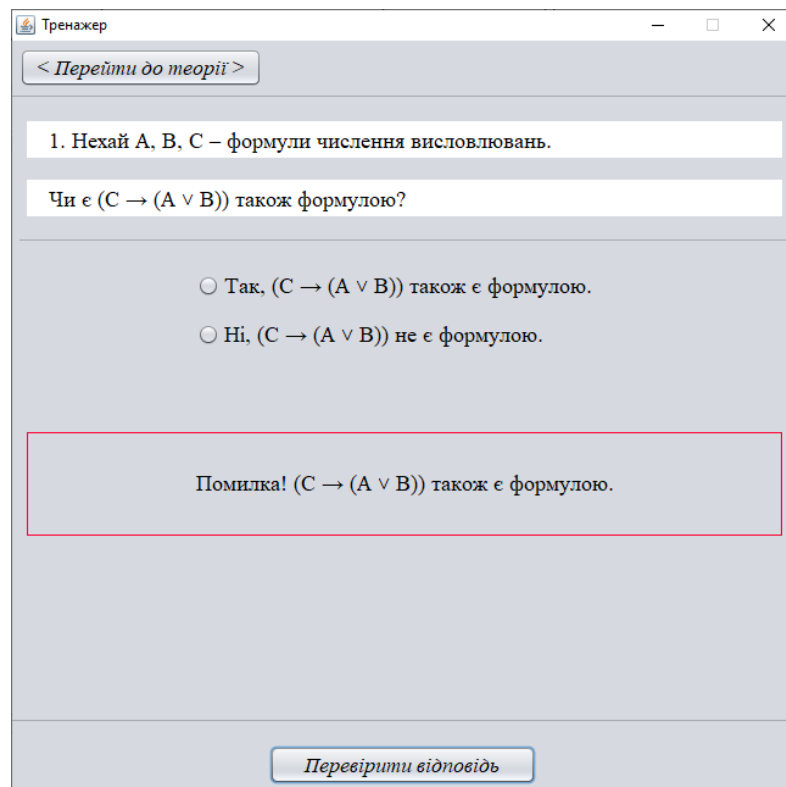


Рисунок 4.11 – Повідомлення про помилку на перше завдання блоку «Формули і підстановки»

В завданні на розбиття дужок на пари у заданій формулі слід вписати ці пари в поля (рис. 4.7).

Останній крок першого блоку – повідомлення про ознайомлення з цієї темою (рис. 4.12).

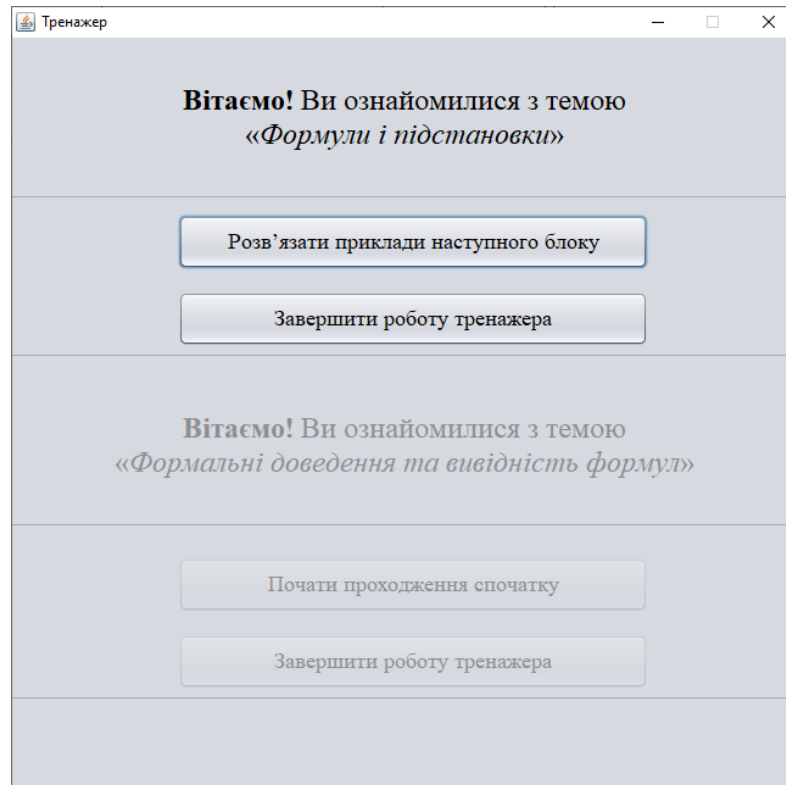


Рисунок 4.12 – Повідомлення про ознайомлення з темою «Формули і підстановки»

Щоб перейти до наступного блоку прикладів потрібно натиснути кнопку «Розв'язати приклади наступного блоку». Для виходу – «Завершити роботу тренажера».

В другому блоці необхідно вказувати послідовність для кожного доведення. Спочатку вказується перший крок (рис. 4.8). Після цього слід натиснути «Перевірити відповідь». Якщо помилка – відобразиться повідомлення про це (рис. 4.9). Для продовження потрібно натиснути «Продовжити».

Розв'язавши всі приклади цього блоку, виведеться повідомлення про ознайомлення з темою «Формальні доведення та вивідність формул». Пропонується почати спочатку або завершити роботу (рис. 4.10).

ВИСНОВКИ

Основні результати роботи:

- розглянуто основні види та призначення платформ дистанційного навчання;
- оглянуто математичі тренажери та їх вади;
- описано актуальність теми роботи;
- розглянуто теоретичні відомості з теми «Методи перевірки тотожної істинності логічних формул»;
- розглянуто специфіку застосування методів перевірки тотожної істинності логічних формул;
- розроблено алгоритм тренажера, що дозволить закріпити знання і навички;
- розроблено тренажер з теми «Методи перевірки тотожної істинності логічних формул»;
- обґрунтовано вибір програмних засобів;
- описано процес програмної реалізації;
- вказано інструкцію по використанню.

До розглянутих тренажерів відносяться тренажери з дисциплін «Математичний аналіз» та «Алгебра і геометрія»:

- тренажер з теми «Розкриття найпростіших невизначеностей»;
- тренажер з теми «Диференціальне числення функції однієї змінної»;
- тренажер з теми "Матриці і визначники";

В алгоритмі описано покроково всі дії при розв'язанні прикладів, що ознайомлять студента з методами перевірки тотожної істинності логічних формул та принципом їх застосування.

При невірній відповіді на завдання слід розробити можливість виведення повідомлення про помилку з правильною відповіддю.

Користувачу на головній сторінці тренажера відображається назва цього тренажера, інформація про розробника і керівника, надається можливість ознайомитися з теоретичним матеріалом і прикладами.

Тренажер містить два блоки прикладів:

1. Формули і підстановки;
2. Формальні доведення та вивідність формул.

Було обрано Java так, як вона дуже популярна мова програмування. З моменту запуску в 1995 році вийшло вже дев'ять основних версій. Java стала технічним феноменом, що багато в чому пов'язано з її унікальною портативністю: Java- додатки працюють на будь-якому пристрої або в операційній системі. Її основні якості – незалежність від архітектури, просте, але надійне виконання програм і гарантія безпеки.

При програмній реалізації першим кроком було розроблено графічне представлення. Другим кроком підключено бібліотеки для подальшого використання різних елементів.

Після оголошення змінних було створено функції, в яких винесено важливі аспекти програми. Так функція `showBlock(int i)` відображає панель з поточним кроком відповідного блоку прикладів. Для виведення необхідного кроку у першому блоці прикладів розроблено `block1ShowStep(int i)`, для другого – `block2ShowStep(int i)`.

Процес виведення повідомлення про помилку реалізовано по різному для кожного блоку прикладів. Так для першого просто змінюється властивість на «показувати» (`setVisible(true)`), а для другого – відбувається перехід між панелями.

Оскільки тренажер має можливість переключатися між українською і англійською мовами, то розроблено функцію `setLanguage()` для зміни всього вмісту тренажера на вибрану мову.

Щоб всі кнопки функціонували потрібним чином на кожную з них створено відповідну подію.

Програму можна використовувати при викладанні іноземним студентам теми «Методи перевірки тотожної істинності логічних формул» для перевірки і закріплення знань, оскільки реалізовано вибір мови між українською та англійською. При початку проходження тренажера блокується вибір мови.

Результати роботи були опубліковані у Збірнику наукових статей магістрів Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання дипломної роботи для студентів ступеня магістра спеціальності 122 «Комп'ютерні науки» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2018. – 35 с.
2. Дистанційне навчання - від теорії до практики, актуальний мережевий семінар // Освітній портал [Електронний ресурс]. – Режим доступу: <http://www.osvita.org.ua/distance/articles/15/>
3. Вишнівський В.В Організація дистанційного навчання. Створення електронних навчальних курсів та електронних тестів. / В.В. Вишнівський, М.П. Гніденко, Г.І. Гайдур, О.О. Ільїн – Навчальний посібник. – Київ: ДУТ, 2014. – 140 с.
4. Дистанційний курс «Математичний аналіз частина 1 (2018)» // Головний центр дистанційного навчання вищого навчального закладу УКООПСІЛКИ «Полтавський університет економіки і торгівлі». – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=282>
5. Макаренко Я.М. Тренажер з теми "Розкриття найпростіших невизначеностей" та розробка його програмного забезпечення з дистанційного навчального курсу "Математичний аналіз" / Я.М. Макаренко // Інформатика та системні науки (ІСН-2014) : матеріали V Всеукр.-наук.-практ. конф., (м. Полтава, 13–15 березня 2014 р.). – Полтава: ПУЕТ, 2014. – С. 204-206. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/2837>
6. Дистанційний курс «Алгебра і геометрія» // Головний центр дистанційного навчання вищого навчального закладу УКООПСІЛКИ «Полтавський університет економіки і торгівлі».
7. Нікольський Ю.В. Дискретна математика / Ю.В. Нікольський, В.В. Пасічник, Ю.М. Щербина. – К.: Видавнича група BVH, 2007. – 368 с.
8. Таран Т.А. Основи дискретної математики / Т.А. Таран. – К.: Просвіта, 2003. – 288 с.

9. Жук П. Ф. Математична логіка та теорія алгоритмів : практикум / уклад.: / П. Ф. Жук – К. : НАУ, 2014. – 21 с.
10. Мова програмування Java та платформа JavaFx: приклади застосування // Державний Університет Телекомунікацій [Електронний ресурс]. – Режим доступу: <http://www.dut.edu.ua/ua/news-1-626-6031-mova-programuvannya-java-ta-platforma-javafx-prikladi-zastosuvannya>
11. Вісім цікавих фактів про Java // Ехо Актуальні новини та цікаві статті зі світу ІТ для програмістів і всіх, хто пов'язаний з розробкою програмного забезпечення. [Електронний ресурс]. – Режим доступу: <https://echo.lviv.ua/dev/2907>
12. Java applets // Вікі ЦДПУ [Електронний ресурс]. – Режим доступу: https://wiki.cuspu.edu.ua/index.php/Java_applets
13. NetBeans // Матеріал з Вікіпедії — вільної енциклопедії [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/NetBeans>
14. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.